US 20150113239A1

(19) United States

(12) Patent Application Publication

Katori et al.

(10) Pub. No.: US 2015/0113239 A1

(43) Pub. Date: Apr. 23, 2015

(54) CREATION AND MANAGEMENT OF LOGICAL VOLUME SNAPSHOTS UNDER HIERARCHICAL STORAGE SYSTEM

(71) Applicant: **International Business Machines Corporation**, Armonk, NY (US)

(72) Inventors: **Katsuyoshi Katori**, Kanagawa (JP); **Koichi Masuda**, Kanagawa-ken (JP); **Takeshi Nohta**, Ibaragi pref. (JP)

(73) Assignee: **International Business Machines Corporation**, Armonk, NY (US)

(21) Appl. No.: **14/516,815**

(22) Filed: **Oct. 17, 2014**

Related U.S. Application Data

(63) Continuation of application No. 14/488,548, filed on Sep. 17, 2014.

(30) **Foreign Application Priority Data**

Oct. 18, 2013 (JP) 2013-216977

Publication Classification

(51) **Int. Cl.**
G06F 3/06 (2006.01)

(52) **U.S. Cl.**
CPC *G06F 3/0665* (2013.01); *G06F 3/065* (2013.01); *G06F 3/0619* (2013.01); *G06F 3/068* (2013.01)

(57) **ABSTRACT**

The method of the present invention includes the steps of: receiving a command for creating a snapshot of the logical volume; preparing a management table for managing the snapshot; detecting the storage status of the logical volume; and updating the management table without recalling the logical volume from the second level storage to the first level storage when the logical volume has migrated from the first level storage to the second level storage. After the update, the storage table indicates whether creation of a snapshot is required, whether a logical volume has migrated, and whether the second level storage is the storage location of the logical volume.

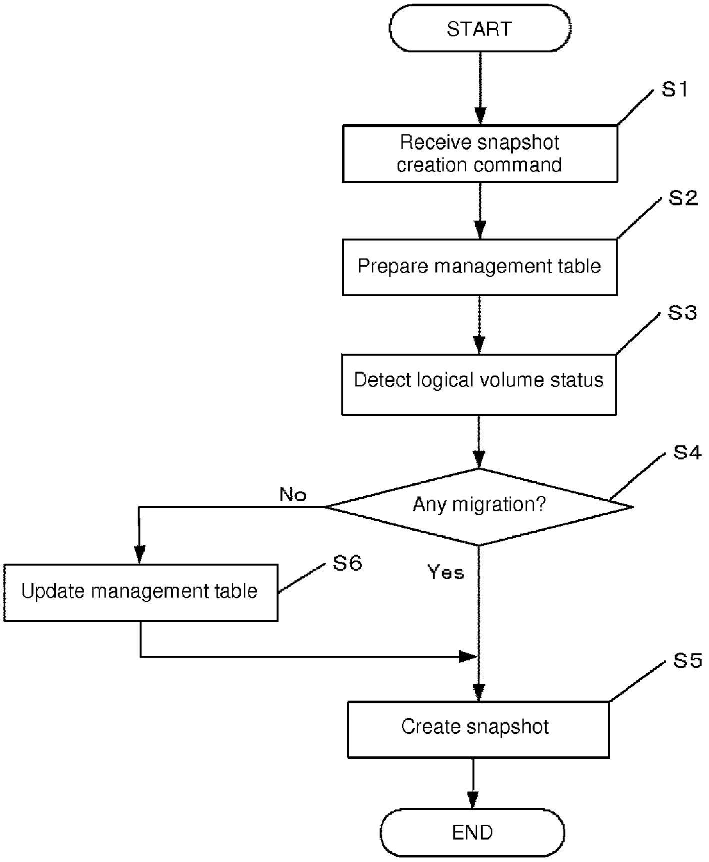

FIG. 1

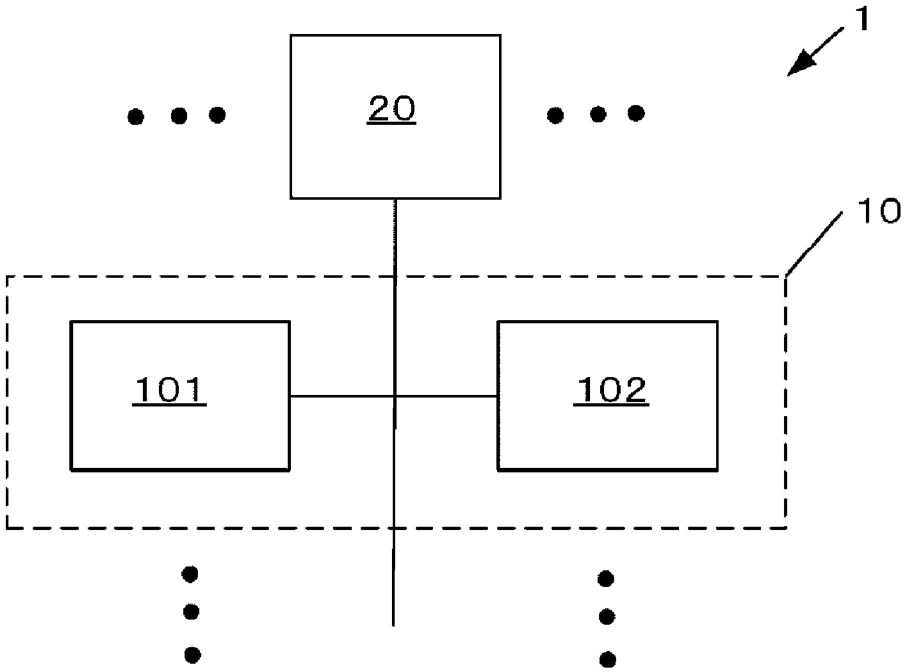

FIG. 2

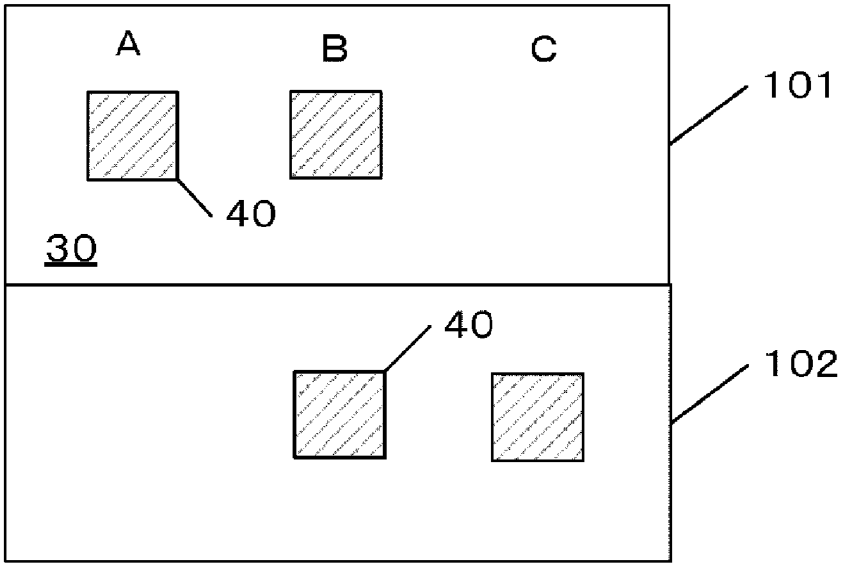

FIG. 3

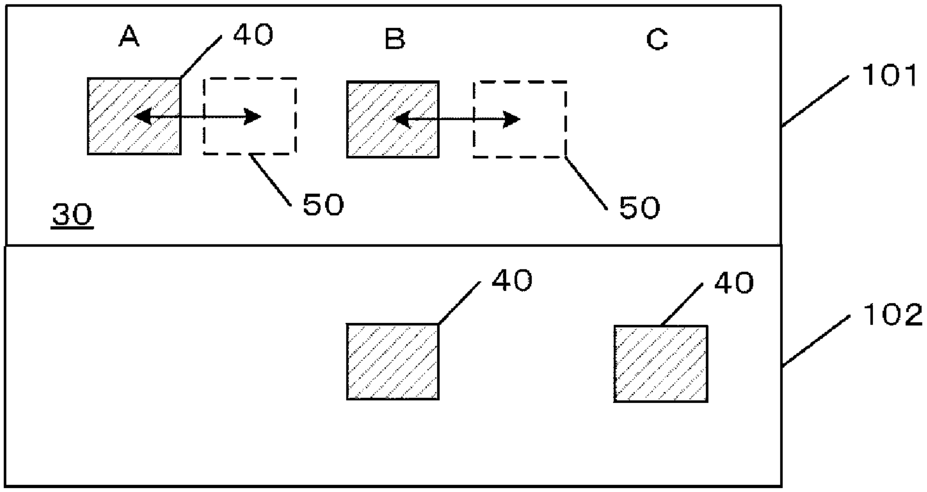

FIG. 4

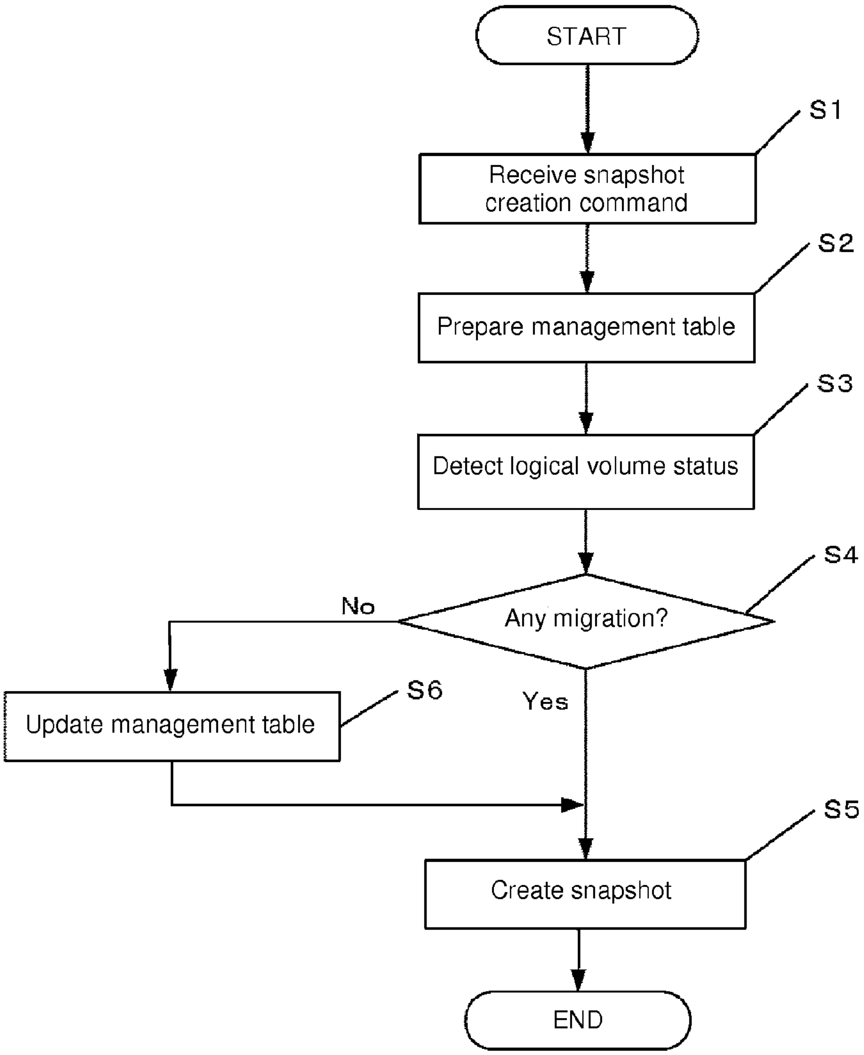
START
S1
Receive snapshot creation command
S2
Prepare management table
S3
Detect logical volume status
S4
Any migration?
No
Yes
S6
Update management table
S5
Create snapshot
END

FIG. 5

LVOL_TOK	snap creation not required
LVOL	migrated
LVOL_TO_PVOL	PVOLX
LVOL_TOK_SNAP	not present (NA)
LVOL_SNAP	not present (NA)
LVOL_TO_PVOL_SNAP	not present (NA)

FIG. 6

LVOL_TOK	snap creation required
LVOL	migrated
LVOL_TO_PVOL	PVOLX
LVOL_TOK_SNAP	snap creation required
LVOL_SNAP	migrated
LVOL_TO_PVOL_SNAP	PVOLX

FIG. 7

LVOL_TOK	snap created
LVOL	pre-migrated/cached
LVOL_TO_PVOL	PVOLX/NA (not present)
LVOL_TOK_SNAP	snap created
LVOL_SNAP	pre-migrated/cached
LVOL_TO_PVOL_SNAP	PVOLX/NA (not present)

FIG. 8

LVOL_TOK	snap created
LVOL	pre-migrated
LVOL_TO_PVOL	PVOLX
LVOL_TOK_SNAP	snap created
LVOL_SNAP	pre-migrated
LVOL_TO_PVOL_SNAP	PVOLX

FIG. 9

LVOL_TOK	snap creation not required
LVOL	cached
LVOL_TO_PVOL	deleted (NA)
LVOL_TOK_SNAP	snap creation not required
LVOL_SNAP	migrated
LVOL_TO_PVOL_SNAP	PVOLX

FIG. 10

LVOL_TOK	deleted (NA)
LVOL	deleted (NA)
LVOL_TO_PVOL	deleted (NA)
LVOL_TOK_SNAP	snap creation not required
LVOL_SNAP	migrated
LVOL_TO_PVOL_SNAP	PVOLX

FIG. 11

LVOL_TOK	snap creation not required
LVOL	migrated
LVOL_TO_PVOL	PVOLX
LVOL_TOK_SNAP	snap creation not required
LVOL_SNAP	cached
LVOL_TO_PVOL_SNAP	deleted (NA)

FIG. 12

LVOL_TOK	snap creation not required
LVOL	migrated
LVOL_TO_PVOL	PVOLX
LVOL_TOK_SNAP	deleted (NA)
LVOL_SNAP	deleted (NA)
LVOL_TO_PVOL_SNAP	deleted (NA)

CREATION AND MANAGEMENT OF LOGICAL VOLUME SNAPSHOTS UNDER HIERARCHICAL STORAGE SYSTEM

[0001] The present application is a continuation nonprovisional application claiming the priority of the filing date of the co-pending and commonly assigned U.S. patent application Ser. No. 14/488,548 (Attorney Docket No. JP920130087US1) entitled "Creation and Management of Logical Volume Snapshots Under Hierarchical Storage System," filed on Sep. 17, 2014.

TECHNICAL FIELD

[0002] The present invention relates to a storage system and, more specifically, to a method for creating and managing hierarchically managed logical volume snapshots in a storage system.

BACKGROUND ART

[0003] In storage systems, data (files) are hierarchically managed using relatively fast, higher-level storage such as a hard disk drive (HDD), and relatively slow, lower-level storage such as a tape drive. In storage systems, logic volume management (LVM) is also performed using logical volumes logically segmented into volume groups of more than one logical volume.

[0004] In copy services providing various types of storage subsystems, file systems and applications in addition to this type of storage management, techniques are used to store images of data at a certain instant (hereinafter referred to as "Time 0"). Examples of these techniques include snapshots and flash copies. The detailed mechanisms of this technique differ depending on the hardware and software being used but pointer information at locations where files (data) are being stored at Time 0 is generally saved, and this pointer information is used to manage the original files and images created at Time 0. In the following explanation, the term "snapshot" is used synonymously with "flash copy".

[0005] Existing snapshots are valid only for files stored on the disk at Time 0, and a snapshot cannot be created when the target file is not present on the disk. In other words, snapshots are of a disk on a single level, and can be created and managed by creating and using pointer data when valid data is present on the disk. A snapshot cannot be created if data is not present on the disk.

[0006] However, when files are hierarchically managed using a disk and physical tape, valid data is sometimes present on the physical tape even when there is no valid data present on the disk. Also, data can be moved from disk to physical tape hierarchically. When files are hierarchically managed in these situations, it is difficult to create and manage snapshots by using the simple pointer information archived by the conventional snapshot functions.

[0007] Patent Literature 1 describes a file management method for hierarchical management of files using a disk and physical tape which includes snapshot files using i-nodes of each source file and i-nodes of each snapshot file for each source file. However, Patent Literature 1 does not describe a method for creating and managing snapshots when no valid data is present on the disk but valid data is present on physical tape.

SUMMARY OF INVENTION

[0008] Therefore, it is an object of the present invention to provide a method for creating and managing snapshots of hierarchically managed logical volumes even in situations in which there is no valid data on a disk but there is valid data on a physical tape.

[0009] The present invention is a method for creating and managing snapshots of hierarchically managed logical volumes in a storage system. The storage system includes first level storage and second level storage lower than the first level storage. This method includes the steps of:

[0010] (a) receiving a command for creating a snapshot of the logical volume;

[0011] (b) preparing a management table for managing the snapshot, the management table including first metadata indicating the creation status of the snapshot, second metadata indicating the management status of the snapshot, and third metadata indicating the storage location of the logical volume;

[0012] (c) detecting the storage status of the logical volume; and

[0013] (d) updating the management table without recalling the logical volume from the second level storage to the first level storage when the logical volume has migrated from the first level storage to the second level storage, the first metadata indicating that creation of a snapshot is required, the second metadata indicating that the logical volume has migrated, and the third metadata indicating that the second level storage is the storage location of the logical volume.

[0014] By updating and reusing a management table to manage a snapshot, the present invention is able to create and manage a snapshot of hierarchically managed logical volumes when the logical volume has migrated from the first level storage to the second level storage without directly having to immediately recall the logical volume from the second level storage to the first level storage.

[0015] The method in one aspect of the present invention also includes the steps of:

[0016] (e) recalling the logical volume when access to the logical volume or the snapshot is requested;

[0017] (f) creating a snapshot from the logical volume recalled to the first level storage; and

[0018] (g) updating the management table, the first metadata indicating that the snapshot has been created, the second metadata indicating that the logical volume has pre-migrated, and the third metadata indicating that the second level storage is the storage location of the logical volume.

[0019] By creating a snapshot and updating and reusing a management table when access to a logical volume or snapshot is requested, one aspect of the present invention is the ability to respond to the access request and properly manage the snapshot afterwards.

[0020] The method in one aspect of the present invention also includes the step of:

[0021] (h) updating the management table when overwriting of the logical volume or deletion of the logical volume is requested, the first metadata indicating that creation of a snapshot is not required, the second metadata indicating that the logical volume has migrated, and the third metadata indicating that the second level storage is the storage location of the logical volume.

[0022] By not requiring the creation of a snapshot and keeping the second level storage as the storage location of logical volume, one aspect of the present invention is the ability to create and manage logical volumes and their snapshots stored in this location when the overwriting or deletion of a logical volume previously used in a command to create a snapshot is requested.

[0023] The method in one aspect of the present invention also includes the step of:

[0024] (i) updating the management table when overwriting of the snapshot is requested, the first metadata indicating that creation of a snapshot is not required, the second metadata indicating that the logical volume is located in the first level storage, and the third metadata indicating that deletion has been performed.

[0025] The method in one aspect of the present invention also includes the step of:

[0026] (j) updating the management table when deletion of the snapshot is requested, the first metadata indicating that deletion has been performed, the second metadata indicating that deletion has been performed, and the third metadata indicating that deletion has been performed.

BRIEF DESCRIPTION OF DRAWINGS

[0027] FIG. **1** is a diagram showing a configuration example of a storage system used by the method of the present invention.

[0028] FIG. **2** is a diagram showing an example of a data storage status.

[0029] FIG. **3** is an image diagram showing a snapshot that was created in the situation shown in FIG. **2**.

[0030] FIG. **4** is a flowchart of the method of the present invention.

[0031] FIG. **5** is a diagram showing an example of a management table prepared in Step S2 of FIG. **4**.

[0032] FIG. **6** is a diagram showing the management table after an update.

[0033] FIG. **7** is a diagram showing the management table after an update.

[0034] FIG. **8** is a diagram showing the management table after an update.

[0035] FIG. **9** is a diagram showing the management table after an update.

[0036] FIG. **10** is a diagram showing the management table after an update.

[0037] FIG. **11** is a diagram showing the management table after an update.

[0038] FIG. **12** is a diagram showing the management table after an update.

DETAILED DESCRIPTION

[0039] The following is an explanation of an embodiment of the present invention with reference to the drawings. FIG. **1** is a diagram showing a configurational example of a storage system used by the method of the present invention. The storage system **10** includes first storage **101** and second storage **102** which can communicate so as to enable communication. FIG. **1** shows a configuration with the minimal requirements for hierarchical management. The storage system **10** can be connected to multiple hosts (servers), and can include multiple units of first storage **101** and second storage **102**. A configuration including the storage system **10** and the host **20** can serve as an entire system **1** (for example, a data system, or data management system) or as part of a system.

[0040] In one embodiment, the first storage **101** in the storage system **10** can be storage including a higher level disk (sometimes referred to as the disk cache or HDD below), and the second storage **102** can be storage including lower level physical tape (tape drive). The configuration (or combination) of first and second storage can be any hierarchically organized combination and is not restricted to combinations of disks and physical tape. In the following explanation, the storage system is Virtual Tape Server TS7700® provided by the applicant, International Business Machines Corporation. (hereinafter referred to as the TS7700.) The TS7700 can include, as a minimal configuration, the disk cache and physical tape illustrated in the storage system **10** of FIG. **1**.

[0041] FIG. **2** shows the storage states of data in the TS7700. The data written from the host **20** is written to the disk cache **101** in units known as “logical volumes” **30**. As mentioned above, logical volumes **30** are segmented logically into volume groups composed of more than one physical volume. The host **20** issues a mount command for a logical volume **30**, and the logical volume **30** is read or written. The host **20** then rewinds, unloads, and ends access. After access has been ended, the data is moved from the disk cache **101** to physical tape **102** and hierarchically managed according to the policies established by the user for each logical volume **30**.

[0042] As shown in FIG. **2**, data **40** inside the TS7700 is stored in one of the following three storage states.

[0043] A: Present only in the disk cache **101**

[0044] B: Present both in the disk cache **101** and on physical tape **102**

[0045] C: Present only on physical tape **102**

A transition from storage state A to storage state B is referred to as pre-migration, a transition from storage state B to storage state C is referred to as migration, and transition from storage state C to storage state B is referred to as a recall.

[0046] In the TS7700, data in the disk cache is read and written via the General Parallel File System® (GPFS). The GPFS provides a snapshot function on both the file system level and the individual file level. In the following explanation, the logical volume which is the source of a snapshot is referred to as the original data and the image of the logical volume created at Time 0 is referred to as the snapshot.

[0047] FIG. **3** shows an image when a snapshot has been created at Time 0 using a function provided by GPFS with regard to the storage states A, B and C shown in FIG. **2**. In FIG. **3**, block **50** delineated by dotted lines is a snapshot of data **40**. Because data **40** is present in the disk cache **101** in storage states A and B, it can create a snapshot **50** properly at Time 0.

[0048] Because data is not present in the disk cache **101** in storage state C, a snapshot cannot be created using GPFS even when there is valid (the latest) data **40** present on the physical tape **102**. Therefore, in order to create a snapshot of data in storage state C at Time 0, an operation has to be performed in advance to recall data on the physical tape and write the data to the disk cache **101**.

[0049] However, in the TS7700, a maximum of 2 million logical volumes can be defined. When there is a large amount of migrated logical volumes, it can take a long time to recall all of the logical volumes. Therefore, Time 0 itself cannot be defined. Also, all volumes cannot be recalled at once because of the limited disk cache capacity. Therefore, it is inappropriate and practically impossible to recall data on the physical tape in advance. This particular problem is to be solved by the method of the present invention explained in detail below.

[0050] The following is an explanation with reference to FIG. **4** of the operational flow of the method of the present invention which is used to create and manage a snapshot of hierarchically managed logical volumes. The operational flow in FIG. **4** is embodied in the configuration shown in FIG. **1** by software executed by a computer (controller) in the storage system **10**.

[0051] In Step S**1**, a command is received from the host to create a snapshot of the logical volumes. More specifically, the user issues a command to the storage system via the host or a management interface. In Step S**2**, a management table is prepared for managing the snapshot. More specifically, a DB2 table is prepared (duplicated) for the snapshot. Here, a DB2 table refers to a DB2 table defined by the TS7700 as a relational database. The DB2 table includes various types of metadata used to manage the snapshot as explained below.

[0052] FIG. **5** is an example of a management table (DB2 table) prepared in Step S**2**. In the management table, LVOL_TOK, LVOL and LVOL_TO_PVOL in line 1 through line 3 include metadata information described below that is related to the original data (logical volumes), and LVOL_TOK_SNAP, LVOL_SNAP and LVOL_TO_PVOL_SNAP in line 4 through line 6 include metadata information described below that is related to the snapshot.

[0053] (a) LVOL_TOK: This holds the status of each logical volume (whether or not the most recent data is present, etc.). An entry is created when the user defines a logical volume irrespective of whether the data is located on the disk or the physical tape.

[0054] (b) LVOL: This holds information indicating in which physical volume the logical volume is located (whether or not the data is migrated or pre-migrated). As in the case of LVOL_TOK, an entry is created when the user defines a logical volume irrespective of whether the data is located on the disk or the physical tape.

[0055] (c) LVOL_TO_PVOL: When each logical volume is migrated or pre-migrated, this holds information indicating whether or not the logical volume is located on physical tape. An entry is created when a logical volume has pre-migrated to physical tape.

[0056] In the example shown in FIG. **5**, the management table is for logical volumes (original data) which have already migrated and are present only on physical tape. As a result, "migrated" is indicated for LVOL in line 2, and physical volume PVOLX is indicated as the storage location of the migrated logical volume (original data) for LVOL_TO_PVOL in line 3. LVOL_TOK in line 1 indicates that a snapshot (snap) does not have to be created of the initial state. LVOL_TOK_SNAP, LVOL_SNAP and LVOL_TO_PVOL_SNAP in line 4 through line 6 are not applicable (NA) in the initial state.

[0057] Returning to FIG. **4**, the storage state of the logical volumes (original data) are detected in Step S**3**. More specifically, it is detected whether or not the logical volumes are in state A, B or C shown in FIG. **2**. In Step S**4**, it is determined whether or not the logical volumes (original data) has migrated. If the determination is YES, there is no data present on the disk cache. Therefore, an actual snapshot is not created and an image cannot be obtained at Time 0 even when a command has been issued to the GPFS to create a snapshot. Thus, the management table is updated in Step S**5** without performing the processing required to create a snapshot right away.

[0058] FIG. **6** shows the management table after the update. First, "snap creation required" is recorded in LVOL_TOK and in LVOL_TOK_SNAP, which is the field indicating the current snapshot status. This indicates that another snapshot has to be created for each logical volume unit at a stage when the original data has been recalled. Then, "migrated" is recorded in both LVOL_SNAP and LVOL which indicates that the logical volumes have migrated. The physical volume PVOLX of the migrated original data is recorded in LVOL_TO_PVOL_SNAP.

[0059] When the determination in Step S**4** is NO, original data is present in the disk cache. In Step S**5**, a snapshot is created of the original data at Time 0, as shown in the examples of states A and B in FIG. **3**. The step goes to S**5**, and the management table is updated. FIG. **7** shows the updated management table after a snapshot has been created. In the management table shown in FIG. **7**, "cached" refers to storage state A in FIG. **2** in which original data is present in the disk cache **101**, and "pre-migrated" refers to storage state B in FIG. **2** in which original data is present in both the disk cache **101** and physical tape **102**.

[0060] Time 0 snapshots of all the logical volumes managed by TS7700 have been completed at this point. However, an actual snapshot image is not created at Time 0 of the logical volumes which have migrated. In the present invention, when any of the following events occurs after the Time 0 snapshot, a separate snapshot is created and managed for the migrated logical volumes. In one embodiment, the "mmcrsnapshot" command in GPFS can be used to create a snapshot at once of all of the data in the managed file system. A snapshot of separate files can be created using the "mmclone" command.

[0061] (A) If Original Data or Snapshot has been Accessed (Logical Volumes are Read and Written)

[0062] (A1) Because the host can only access files located in the disk cache, the TS7700 recalls migrated original data when the data has been accessed. When a snapshot is accessed, it is recalled from the physical tape written to LVOL_TO_PVOL_SNAP.

[0063] (A2) Original data can be accessed from the host once it has been recalled and its snapshot is created in an instant.

[0064] (A3) Afterwards, the management table is updated.

[0065] FIG. **8** shows the management table after it has been updated.

[0066] LVOL_TOK and LVOL_TOK_SNAP have been changed from "snap creation required" to "snap created", and LVOL and LVOL_SNAP have been changed to "pre-migrated".

[0067] (B) If Original Files have been Overwritten

[0068] When "Write From BOT (Beginning of Tape)" has occurred, the management table is updated in the following way. LVOL_TOK has to be changed to "snap creation required", LVOL has to be updated to "cached" to indicate that data is present on the disk, and LVOL_TO_PVOL has to be deleted (NA). However, because an entry for the snapshot remains in LVOL_TO_PVOL_SNAP, the logical volumes of the original file present on the physical tape are not invalidated. In this case, there is no longer any need to create a snapshot, and future access is provided to the original file and snapshot as two separate files. FIG. **9** shows the management table after it has been updated.

[0069] (C) If Original Files have been Deleted

[0070] In this case, LVOL_TOK, LVOL and LVOL_TO_PVOL are deleted and the management table is updated accordingly. The status of LVOL_TOK_SNAP, LVOL_SNAP and LVOL_TO_PVOL_SNAP are the same as case (B) above. FIG. **10** shows the management table after it has been updated.

[0071] A snapshot does not have to be created if any of the following has occurred before a snapshot has been created using GPFS.

[0072] (D) If Snapshot Overwrite (Write from BOT) has Occurred

[0073] In this case, the management table is updated in the following way without any recall because a snapshot does not have to be created of the original files. LVOL_TOK and LVOL_TOK_SNAP are changed from "snap creation required" to "snap creation not required", LVOL_SNAP is updated to "cached" indicating that data is present on the disk, and LVOL_TO_PVOL_SNAP is deleted (NA). FIG. **11** shows the management table after it has been updated.

[0074] (E) If Snapshot Deletion has Occurred

[0075] In this case, the management table is updated in the following way. LVOL_TOK_SNAP, LVOL_SNAP and LVOL_TO_PVOL_SNAP are deleted (NA), and LVOL_TOK is changed from "snap creation required" to "snap creation not required". FIG. **12** shows the management table after it has been updated.

[0076] As mentioned above, three values can be applied to the fields indicating the "status of a current snapshot" added to LVOL_TOK and LVOL_TOK_SNAP: "snap creation not required", "snap creation required", and "snap created". The logical volume and snapshot of "snap creation not required" are treated as separate logical volumes.

[0077] Embodiments of the present invention have been explained with reference to the drawings, but the present invention is not limited to these embodiments. In the embodiments described above, hierarchical management was explained between a disk and physical tape. However, the present invention is not limited to this and can be applied to any system that is hierarchically managed using different levels. A similar mechanism can be realized by applying an existing snapshot function to an upper-level disk device and recording the state of data (metadata) which has been moved to a lower-level disk drive as metadata. Snapshot management can be performed using the same technique even when three or more levels are used.

REFERENCE SIGNS LIST

[0078] **1**: Overall system

[0079] **10**: Storage system

[0080] **20**: Host, multiple hosts (servers)

[0081] **30**: Logical volume

[0082] **40**: Data

[0083] **50**: Snapshot

[0084] **101**: 1st storage medium (HDD, etc.)

[0085] **102**: 2nd storage medium (tape drive, etc.)

1. A method for creating and managing snapshots of hierarchically managed logical volumes in a storage system including first level storage and second level storage lower than the first level storage, the method comprising:

receiving a command for creating a snapshot of the logical volume;

preparing, using a processor and a memory, a management table for managing the snapshot, the management table including first metadata indicating the creation status of the snapshot, second metadata indicating the management status of the snapshot, and third metadata indicating the storage location of the logical volume;

detecting the storage status of the logical volume; and

updating the management table without recalling the logical volume from the second level storage to the first level storage when the logical volume has migrated from the first level storage to the second level storage, the first metadata indicating that creation of a snapshot is required, the second metadata indicating that the logical volume has migrated, and the third metadata indicating that the second level storage is the storage location of the logical volume.

2. The method according to claim **1** further comprising:

recalling the logical volume when access to the logical volume or the snapshot is requested;

creating a snapshot from the logical volume recalled to the first level storage; and

updating the management table, the first metadata indicating that the snapshot has been created, the second metadata indicating that the logical volume has pre-migrated, and the third metadata indicating that the second level storage is the storage location of the logical volume.

3. The method according to claim **1** further comprising:

updating the management table when overwriting of the logical volume or deletion of the logical volume is requested, the first metadata indicating that creation of a snapshot is not required, the second metadata indicating that the logical volume has migrated, and the third metadata indicating that the second level storage is the storage location of the logical volume.

4. The method according to claim **1** further comprising:

updating the management table when overwriting of the snapshot is requested, the first metadata indicating that creation of a snapshot is not required, the second metadata indicating that the logical volume is located in the first level storage, and the third metadata indicating that deletion has been performed.

5. The method according to claim **1** further comprising:

updating the management table when deletion of the snapshot is requested, the first metadata indicating that deletion has been performed, the second metadata indicating that deletion has been performed, and the third metadata indicating that deletion has been performed.

6. The method according to claim **1** further comprising:

creating a snapshot of the logical volume in the first level storage when the logical volume is stored in the first level storage or stored in both the first level storage and the second level storage; and

updating the management table, the first metadata indicating that the snapshot has been created, the second metadata indicating that the logical volume is stored in the first level storage or is stored in both the first level storage and the second level storage, and the third metadata indicating that the second level storage is not the storage location of the logical volume or the second level storage is the storage location of the logical volume.

7. The method according to claim **1**, wherein the first level storage includes a hard disk drive (HDD), and the second level storage includes a tape drive.

* * * * *