Description

METHOD, SYSTEM, AND PROGRAM FOR INCREMENTAL VIRTUAL COPY

Technical Field

[001] The present invention is directed to incremental virtual copy.

Background Art

[002] Computing systems often include one or more host computers ("hosts") for processing data and running application programs, direct access storage devices (DASDs) for storing data, and a storage controller for controlling the transfer of data between the hosts and the DASD. Storage controllers, also referred to as control units or storage directors, manage access to a storage space comprised of numerous hard disk drives connected in a loop architecture, otherwise referred to as a Direct Access Storage Device (DASD). Hosts may communicate Input/Output (I/O) requests to the storage space through the storage controller.

[003] In many systems, data on one storage device, such as a DASD, may be copied to the same or another storage device so that access to data volumes can be provided from two different devices. A point-in-time copy involves physically copying all the data from source volumes to target volumes so that the target volume has a copy of the data as of a point-in-time. A point-in-time copy can also be made by logically making a copy of the data and then only copying data over when necessary, in effect deferring the physical copying. This logical copy operation is performed to minimize the time during which the target and source volumes are inaccessible.

[004] A number of direct access storage device (DASD) subsystems are capable of performing "instant virtual copy" operations, also referred to as "fast replicate functions." Instant virtual copy operations work by modifying metadata such as relationship tables or pointers to treat a source data object as both the original and copy. In response to a host=s copy request, the storage subsystem immediately reports creation of the copy without having made any physical copy of the data. Only a "virtual" copy has been created, and the absence of an additional physical copy is completely unknown to the host.

[005] Later, when the storage system receives updates to the original or copy, the updates are stored separately and cross-referenced to the updated data object only. At this point, the original and copy data objects begin to diverge. The initial benefit is that the instant virtual copy occurs almost instantaneously, completing much faster than a normal physical copy operation. This frees the host and storage subsystem to perform other tasks. The host or storage subsystem may even proceed to create an actual,

physical copy of the original data object during background processing, or at another time.

[006] One such instant virtual copy operation is known as a FlashCopy7 operation. A FlashCopy7 operation involves establishing a logical point-in-time relationship between source and target volumes on the same or different devices. The FlashCopy7 operation guarantees that until a track in a FlashCopy7 relationship has been hardened to its location on the target disk, the track resides on the source disk. A relationship table is used to maintain information on all existing FlashCopy7 relationships in the subsystem. During the establish phase of a FlashCopy7 relationship, one entry is recorded in the source and target relationship tables for the source and target that participate in the FlashCopy7 being established. Each added entry maintains all the required information concerning the FlashCopy7 relationship. Both entries for the relationship are removed from the relationship tables when all FlashCopy7 tracks from the source extent have been physically copied to the target extents or when a withdraw command is received. In certain cases, even though all tracks have been copied from the source extent to the target extent, the relationship persists.

[007] The target relationship table further includes a bitmap that identifies which tracks involved in the FlashCopy7 relationship have not yet been copied over and are thus protected tracks. Each track in the target device is represented by one bit in the bitmap. The target bit is set when the corresponding track is established as a target track of a FlashCopy7 relationship. The target bit is reset when the corresponding track has been copied from the source location and destaged to the target device due to writes on the source or the target device, or a background copy task.

[008] In the prior art, as part of the establishment of the logical point-in-time relationship during the FlashCopy7 operation, all tracks in the source cache that are included in the FlashCopy7 relationship must be destaged to the physical source volume, e.g., source DASD, and all tracks in the target cache included in the FlashCopy7 must be discarded.

[009] Once the logical relationship is established, hosts may then have immediate access to data on the source and target volumes, and the data may be copied as part of a background operation. A read to a track that is a target in a FlashCopy7 relationship and not in cache triggers a stage intercept, which causes the source track corresponding to the requested target track to be staged to the target cache when the source track has not yet been copied over and before access is provided to the track from the target cache. This ensures that the target has the copy from the source that existed at the point-in-time of the FlashCopy7 operation. Further, any destages to tracks on the source device that have not been copied over triggers a destage intercept, which causes the tracks on the source device to be copied to the target device.

[010] Instant virtual copy techniques have been developed, at least in part, to quickly create a duplicate copy of data without interrupting or slowing foreground processes. Instant virtual copy techniques, such as a FlashCopy7 operation, provide a point-in-time copy tool. Instant virtual copy techniques may be used for a variety of applications, including, for example, data backup, data migration, data mining, testing, etc. For example, an instant virtual copy technique may be used for the creation of a physical "backup" copy of the source data, to aid in disaster recovery.

[011] Although the instant virtual copy techniques are useful for copying large amounts of data, conventional instant virtual copy techniques may be improved. In particular, there is a need in the art for improved instant virtual copy techniques that avoid physically copying large amounts of data.

Disclosure of Invention

[012] In a first aspect, the present invention provides a method for reducing an amount of data transferred, comprising: maintaining a first indicator for each source block of data to indicate whether the source block of data has been updated in source storage since the source block of data was last transferred to target storage; maintaining a second indicator for each target block of data in the target storage to indicate whether the target block of data has been updated in the target storage since the target block of data was overwritten by a corresponding source block of data; and when transferring data from the source storage to the target storage, transferring each source block of data for which a first indicator has been set to indicate that the source block of data has been updated; and transferring each source block of data that corresponds to a target block of data for which a second indicator has been set to indicate that the target block of data has been updated.

[013] The method preferably further comprises: maintaining a third indicator for each target block of data to indicate whether a corresponding source block of data or the target block of data is to be retrieved for a stage operation.

[014] The method preferably further comprises: when a source block of data has been copied to the target storage, updating the third indicator for the target block of data.

[015] The method preferably further comprises: receiving an incremental virtual copy operation; and updating the third indicator for each target block of data based on the first indicator for the source block of data and the second indicator for the target block of data corresponding to the source block of data, wherein transferring data from the source storage to the target storage comprises transferring each source block of data for which a third indicator indicates that the source block of data is to be transferred to the target storage.

[016] The method preferably further comprises: after establishing an incremental virtual

copy relationship, updating the third indicator for each target block of data to indicate that the source block of data corresponding to the target block of data is to be retrieved from the source storage for a stage operation.

[017] The method preferably further comprises: after transferring data from the source storage to the target storage, updating the first indicator for each source block of data to indicate that the source block of data has not been updated in the source storage since the source block of data was last transferred to the target storage.

[018] The method preferably further comprises: after transferring data from the source storage to the target storage, updating the second indicator for each target block of data to indicate that the target block of data has not been updated in the target storage since the source block of data was last transferred to the target storage.

[019] Preferably, each source block of data and each target block of data comprises a track of a volume.

[020] Preferably, each source block of data and each target block of data comprises a sector of a volume.

[021] The method preferably further comprises: creating an incremental virtual copy relationship between a source portion of data and a target portion of data by performing an incremental virtual copy operation between the source portion of data and the target portion of data, wherein the first indicator and the second indicator are maintained for each source block of data and target block of data in the source and target portions of data in the incremental virtual copy relationship, respectively.

[022] The method preferably further comprises: updating the first indicator for each source block of data to indicate that the source block of data has not been updated; updating the second indicator for each target block of data to indicate that the target block of data has not been updated; and updating a third indicator for each target block of data to indicate that the source block of data needs to be transferred to the target storage.

[023] The method preferably further comprises: when a source block of data is updated, updating the first indicator for the source block of data to indicate that the source block of data has been updated.

[024] The method preferably further comprises: when a target block of data is updated, updating the second indicator for the target block of data to indicate that the target block of data has been updated.

[025] There may also be provided an article of manufacture for reducing an amount of data transferred, wherein the article of manufacture causes operations, the operations comprising: maintaining a first indicator for each source block of data to indicate whether the source block of data has been updated in source storage since the source block of data was last transferred to target storage; maintaining a second indicator for

each target block of data in the target storage to indicate whether the target block of data has been updated in the target storage since the target block of data was overwritten by a corresponding source block of data; and when transferring data from the source storage to the target storage, transferring each source block of data for which a first indicator has been set to indicate that the source block of data has been updated; and transferring each source block of data that corresponds to a target block of data for which a second indicator has been set to indicate that the target block of data has been updated.

[026] The article of manufacture preferably comprises operations further comprising: maintaining a third indicator for each target block of data to indicate whether a corresponding source block of data or the target block of data is to be retrieved for a stage operation.

[027] The article of manufacture preferably comprises: when a source block of data has been copied to the target storage, updating the third indicator for the target block of data.

[028] The article of manufacture preferably comprises: receiving an incremental virtual copy operation; and updating the third indicator for each target block of data based on the first indicator for the source block of data and the second indicator for the target block of data corresponding to the source block of data, wherein transferring data from the source storage to the target storage comprises transferring each source block of data for which a third indicator indicates that the source block of data is to be transferred to the target storage.

[029] The article of manufacture preferably comprises: after establishing an incremental virtual copy relationship, updating the third indicator for each target block of data to indicate that the source block of data corresponding to the target block of data is to be retrieved from the source storage for a stage operation.

[030] The article of manufacture preferably comprises: after transferring data from the source storage to the target storage, updating the first indicator for each source block of data to indicate that the source block of data has not been updated in the source storage since the source block of data was last transferred to the target storage.

[031] The article of manufacture preferably comprises: after transferring data from the source storage to the target storage, updating the second indicator for each target block of data to indicate that the target block of data has not been updated in the target storage since the source block of data was last transferred to the target storage.

[032] Preferably each source block of data and each target block of data comprises a track of a volume.

[033] Preferably, each source block of data and each target block of data comprises a sector of a volume.

[034] The article of manufacture preferably comprises: creating an incremental virtual copy relationship between a source portion of data and a target portion of data by performing an incremental virtual copy operation between the source portion of data and the target portion of data, wherein the first indicator and the second indicator are maintained for each source block of data and target block of data in the source and target portions of data in the incremental virtual copy relationship, respectively.

[035] The article of manufacture preferably comprises: updating the first indicator for each source block of data to indicate that the source block of data has not been updated; updating the second indicator for each target block of data to indicate that the target block of data has not been updated; and updating a third indicator for each target block of data to indicate that the source block of data needs to be transferred to the target storage.

[036] The article of manufacture preferably comprises: when a source block of data is updated, updating the first indicator for the source block of data to indicate that the source block of data has been updated.

[037] The article of manufacture preferably comprises: when a target block of data is updated, updating the second indicator for the target block of data to indicate that the target block of data has been updated.

[038] In a second aspect, the present invention provides a system for reducing an amount of data transferred, comprising: means for maintaining a first indicator for each source block of data to indicate whether the source block of data has been updated in source storage since the source block of data was last transferred to target storage; means for maintaining a second indicator for each target block of data in the target storage to indicate whether the target block of data has been updated in the target storage since the target block of data was overwritten by a corresponding source block of data; and when transferring data from the source storage to the target storage, means for transferring each source block of data for which a first indicator has been set to indicate that the source block of data has been updated; and means for transferring each source block of data that corresponds to a target block of data for which a second indicator has been set to indicate that the target block of data has been updated.

[039] The system preferably comprises: means for maintaining a third indicator for each target block of data to indicate whether a corresponding source block of data or the target block of data is to be retrieved for a stage operation.

[040] The system preferably comprises: when a source block of data has been copied to the target storage, means for updating the third indicator for the target block of data.

[041] The system of preferably comprises: means for receiving an incremental virtual copy operation; and means for updating the third indicator for each target block of data based on the first indicator for the source block of data and the second indicator for the

target block of data corresponding to the source block of data, wherein transferring data from the source storage to the target storage comprises transferring each source block of data for which a third indicator indicates that the source block of data is to be transferred to the target storage.

[042] The system preferably comprises: after establishing an incremental virtual copy relationship, means for updating the third indicator for each target block of data to indicate that the source block of data corresponding to the target block of data is to be retrieved from the source storage for a stage operation.

[043] The system preferably comprises: after transferring data from the source storage to the target storage, means for updating the first indicator for each source block of data to indicate that the source block of data has not been updated in the source storage since the source block of data was last transferred to the target storage.

[044] The system preferably comprises: after transferring data from the source storage to the target storage, means for updating the second indicator for each target block of data to indicate that the target block of data has not been updated in the target storage since the source block of data was last transferred to the target storage.

[045] Preferably, each source block of data and each target block of data comprises a track of a volume.

[046] Preferably, each source block of data and each target block of data comprises a sector of a volume.

[047] The system preferably comprises: means for creating an incremental virtual copy relationship between a source portion of data and a target portion of data by performing an incremental virtual copy operation between the source portion of data and the target portion of data, wherein the first indicator and the second indicator are maintained for each source block of data and target block of data in the source and target portions of data in the incremental virtual copy relationship, respectively.

[048] The system preferably comprises: means for updating the first indicator for each source block of data to indicate that the source block of data has not been updated; means for updating the second indicator for each target block of data to indicate that the target block of data has not been updated; and means for updating a third indicator for each target block of data to indicate that the source block of data needs to be transferred to the target storage.

[049] The system preferably comprises: when a source block of data is updated, means for updating the first indicator for the source block of data to indicate that the source block of data has been updated.

[050] The system preferably comprises: when a target block of data is updated, means for updating the second indicator for the target block of data to indicate that the target block of data has been updated.

[051] In a third aspect, the present invention provides a computer program comprising computer program code to, when loaded into a computer system and executed thereon, cause said compter system to perform the steps of a method according to the first aspect. Preferred features of the computer program comprise computer program code corresponding to the preferred method steps of the first aspect.

[052] Thus preferably provided are a method, system, and program for reducing an amount of data transferred. A first indicator is maintained for each source block of data to indicate whether the source block of data has been updated in source storage since the source block of data was last transferred to target storage. A second indicator is maintained for each target block of data in target storage to indicate whether the target block of data has been updated in target storage since the target block of data was overwritten by a corresponding source block of data. When transferring data from the source storage to the target storage, each source block of data for which a first indicator has been set to indicate that the source block of data has been updated is transferred and each source block of data that corresponds to a target block of data for which a second indicator has been set to indicate that the target block of data has been updated is transferred.

[053] The described implementations of the invention provide a method, system, and program for creating an incremental virtual copy.

Brief Description of the Drawings

[054] A preferred embodiment of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

[055] FIGs. 1A and 1B illustrate, in block diagrams, a computing environment in accordance with certain implementations of the invention.

[056] FIG. 2 illustrates various structures in accordance with certain implementations of the invention.

[057] FIG. 3 illustrates logic for updating structures in accordance with certain implementations of the invention.

[058] FIG. 4 illustrates logic for performing an incremental virtual copy operation in accordance with certain implementations of the invention.

[059] FIG. 5 illustrates logic implemented in write process for processing of a write operation in accordance with certain implementations of the invention.

[060] FIG. 6 illustrates logic implemented in the read process for processing of a read operation in accordance with certain implementations of the invention

[061] FIG. 7 illustrates a background copy process in accordance with certain implementations of the invention.

[062] FIG. 8 illustrates an architecture of a computer system that may be used in

accordance with certain implementations of the invention.

Mode for the Invention

[063] In the following description, reference is made to the accompanying drawings which form a part hereof and which illustrate several implementations of the present invention. It is understood that other implementations may be utilized and structural and operational changes may be made without departing from the scope of the present invention.

[064] Implementations of the invention provide an incremental virtual copy operation that is an enhancement to an instant virtual copy operation. With the incremental virtual copy operation, only the blocks of data that were updated on source and target volumes since the last instant virtual copy operation from the source volume to the target volume are copied. An incremental virtual copy operation reduces the duration of creating a physical copy of a source volume and minimizes the impact to other applications (e.g., minimizes the usage of bandwidth to physical storage for destaging data, thus allowing more bandwidth for reads from physical storage).

[065] FIGs. 1A and 1B illustrate, in block diagrams, a computing environment in accordance with certain implementations of the invention. A storage controller 100 receives Input/Output (I/O) requests from hosts 140a,b,...l (wherein a, b, and l may be any integer value) over a network 190 directed toward storage devices 120, 130 configured to have volumes (e.g., Logical Unit Numbers, Logical Devices, etc.) 122a,b...n and 132a,b...m, respectively, where m and n may be different integer values or the same integer value. In certain implementations, the size of the target storage 130 may be larger than or equal to the source storage 120.

[066] The source storage 120 includes one or more volumes 122a,b...n, which may be divided into blocks of storage 150 containing blocks of data, and the blocks of storage 150 are further divided into sub-blocks of storage (150a-150p, where a and p may be any integer value) that contain sub-blocks of data. A volume may be any logical or physical element of storage. In certain implementations, the blocks of data are contents of tracks, while the sub-blocks of data are contents of sectors of tracks.

[067] Target storage 130 maintains copies of all or a subset of the volumes 122a,b...n of the source storage 120. Additionally, target storage 130 may be modified by, for example, host 140. Target storage 130 includes one or more volumes 132a,b...m, which may be divided into blocks of storage 150 containing blocks of data, and the blocks of storage 150 are further divided into sub-blocks of storage (150a-150p, where a and p may be any integer value) that contain sub-blocks of data. A volume may be any logical or physical element of storage. In certain implementations, the blocks of data are tracks, while the sub-blocks of data are sectors of tracks.

[068] For ease of reference, the terms tracks and sectors will be used herein as examples of blocks of data and sub-blocks of data, but use of these terms is not meant to limit implementations of the invention to tracks and sectors. The implementations of the invention are applicable to any type of storage, block of storage or block of data divided in any manner. Moreover, although implementations of the invention refer to blocks of data, alternate implementations of the invention are applicable to sub-blocks of data.

[069] The storage controller 100 includes a source cache 124 in which updates to tracks in the source storage 120 are maintained until written to source storage 120 (i.e., the tracks are destaged to physical storage). The storage controller 100 includes a target cache 134 in which updates to tracks in the target storage 130 are maintained until written to target storage 130 (i.e., the tracks are destaged to physical storage). The source cache 124 and target cache 134 may comprise separate memory devices or different sections of a same memory device. The source cache 124 and target cache 134 are used to buffer read and write data being transmitted between the hosts 140a,b...l, source storage 120, and target storage 130. Further, although caches 124 and 134 are referred to as source and target caches, respectively, for holding source or target blocks of data in a point-in-time copy relationship, the caches 124 and 134 may store at the same time source and target blocks of data in different point-in-copy relationships.

[070] Additionally, the storage controller 100 includes a nonvolatile cache 118. The nonvolatile cache 118 may be, for example, a batterybacked up volatile memory, to maintain a nonvolatile copy of data updates.

[071] The storage controller 100 further includes system memory 110, which may be implemented in volatile and/or non-volatile devices. The system memory 110 includes a read process 112 for reading data, a write process 114 for writing data, and an incremental virtual copy process 116. The read process 112 executes in system memory 110 to read data from storages 120 and 130 to caches 124 and 134, respectively. The write process 114 executes in system memory 110 to write data from caches 124 and 134 to storages 120 and 130, respectively. The incremental virtual copy process 116 executes in system memory 110 to perform an incremental virtual copy operation that transfers data from source storage 120 to target storage 130. In certain implementations of the invention, there may be multiple incremental virtual copy processes. In certain implementations of the invention, the incremental virtual copy process may be executed at another storage controller connected to storage controller 100 instead of, or in addition to, execution at the storage controller 100. The system memory 110 may be in a separate memory devices from caches 124 and 134 or may share a memory device with one or both caches 124 and 134.

[072] Implementations of the invention are applicable to the transfer of data between any two storage mediums, which for ease of reference will be referred to herein as source storage and target storage. For example, certain implementations of the invention may be used with two storage mediums located at a single storage controller, as illustrated in FIG. 1A. Moreover, certain alternative implementations of the invention may be used with two storage mediums located at different storage controllers, different physical sites, etc. Also, for ease of reference, a block of data in source storage will be referred to as a "source block of data," and a block of data in target storage will be referred to as a "target block of data."

[073] In certain implementations, removable storage (instead of or in addition to target storage 130) may be used to maintain copies of all or a subset of the source storage 120, and the implementations of the invention transfer data to the removable storage rather than to the target storage. The removable storage may reside at the storage controller 100.

[074] The storage controller 100 may further include a processor complex (not shown) and may comprise any storage controller or server known in the art, such as an Enterprise Storage Server7 (ESS), 39907 Storage Controller, etc. The hosts 140a,b...l may comprise any computing device known in the art, such as a server, mainframe, workstatation, personal computer, hand held computer, laptop telephony device, network appliance, etc. The storage controller 100 and host system(s) 140a,b...l communicate via a network 190, which may comprise a Storage Area Network (SAN), a Local Area Network (LAN), Wide Area Network (WAN), the Internet, an Intranet, etc. The source storage 120 and target storage 130 may each comprise an array of storage devices, such as Direct Access Storage Devices (DASDs), Just a Bunch of Disks (JBOD), Redundant Array of Independent Disks (RAID), virtualization device, etc.

[075] Additionally, although FIG. 1A illustrates a single storage controller, one skilled in the art would know that multiple storage controllers may be connected via a network (e.g., a Local Area Network (LAN), Wide Area Network (WAN), the Internet, etc.), and one or more of the multiple storage controllers may implement the invention.

[076] When host 140 wishes to update a block of data in source storage 120, host 140 writes data to a block of storage in source cache 124. Write operations modify the block of storage in source cache 124 synchronously (i.e., writing host 140 waits for the operation to complete), and then, in a background process, source cache 124 content is written to source storage 120. A write operation may update data, write new data, or write the same data again. Writing data in source cache 124 to source storage 120 is called a destage operation. Copying all or a portion of a block of data from source storage 120 to source cache 124 is a staging operation. Likewise, data may be staged

and destaged between target storage 130 and target cache 134. Moreover, data may be staged from source storage 120 to target cache 134.

[077] FIG. 2 illustrates various structures 200, 210, and 220 in accordance with certain implementations of the invention. Nonvolatile cache 118 includes a target copy structure 200. The target copy structure 200 may be used to determine whether to retrieve data from source storage 120 or target storage 130 to cache 124 or 134, respectively (i.e., for a staging operation). Additionally, the target copy structure 200 may be used to determine which blocks of data in source storage 120 are to be copied to target storage 130. The target copy structure 200 includes an indicator (e.g., a bit) for each block of data in, for example, a volume. When an indicator is set to a first value (e.g., one), the setting indicates that the block of data is to be retrieved from the source storage 120 for a staging operation or indicates that the block of data is to be copied to target storage 130 for an incremental virtual copy operation. When an indicator is set to a second value (e.g., zero), the setting indicates that the block of data is to be retrieved from the target storage 130 for a staging operation or indicates that the block of data is not to be copied from source storage 120 to target storage 130 for an incremental virtual copy operation.

[078] A source change recording structure 210 is used to monitor updates to blocks of data within portions of data in the source storage 120 for which an incremental virtual copy relationship has been established. The source change recording structure 210 includes an indicator (e.g., a bit) for each block of data in the source storage 120 that is part of the incremental virtual copy relationship. When an indicator is set to a first value (e.g., one), the setting indicates that the block of data has been updated since the last incremental virtual copy operation. When an indicator is set to a second value (e.g., zero), the setting indicates that the block of data has not been updated since the last incremental virtual copy operation.

[079] A target change recording structure 220 is used to monitor updates to blocks of data in the target storage 130 after an incremental virtual copy relationship has been established. The target change recording structure 220 includes an indicator (e.g., a bit) for each block of data in the target storage 130 that is part of the incremental virtual copy relationship. When an indicator is set to a first value (e.g., one), the setting indicates that the block of data has been updated since the last incremental virtual copy operation. When an indicator is set to a second value (e.g., zero), the setting indicates that the block of data has not been updated since the last incremental virtual copy operation.

[080] In certain implementations of the invention, each structure 200, 210, and 220 comprises a bitmap, and each indicator comprises a bit. In each structure 200, 210, and 220, the nth indicator corresponds to the nth block of data (e.g., the first indicator in

each structure 200, 210, and 220 corresponds to the first data block). Although the structures 200, 210, and 220 have been illustrated as three separate structures, the structures may be combined in any form without departing from the scope of the invention. In certain implementations of the invention, there is a copy of each structure for each volume. In certain alternative implementations of the invention, there is a single copy of each structure for all volumes.

[081] FIG. 3 illustrates logic implemented in the incremental virtual copy process 116 for updating structures in accordance with certain implementations of the invention. Control begins at block 300 with initial establishment of an incremental virtual copy relationship. The incremental virtual copy relationship is formed between one or more portions of data (e.g., source volumes) in the source storage 120 and corresponding portions of data (e.g., target volumes) in the target storage 130 when an incremental virtual copy operation is performed between the corresponding portions of data. The first incremental virtual copy operation may copy, for example, one or more source volumes to corresponding target volumes. Subsequent copies, however, may make incremental copies, avoiding re-copying any portions of source volumes that have not changed since the last instant virtual copy operation.

[082] In block 310, the incremental virtual copy process 116 updates indicators in the target copy structure 200 to indicate that all of the blocks of data corresponding to the indicators are to be retrieved from source storage for a staging operation and to indicate that all blocks of data are to be copied from source storage to target storage for an incremental virtual copy operation or a physical copy operation. In certain implementations of the invention, the indicators in the target copy structure 200 are set to one.

[083] In block 320, the incremental virtual copy process 116 updates the indicators in the source change recording structure 210 to indicate that the source blocks of data corresponding to the indicators have not been updated since the last incremental virtual copy operation. In certain implementations of the invention, all of the indicators in the source change recording structure 210 are set to zero. In block 330, the incremental virtual copy process 116 updates the indicators in the target change recording structure 220 to indicate that the target blocks of data corresponding to the indicators have not been updated since the last incremental virtual copy operation. In certain implementations of the invention, all of the indicators in the target change recording structure 220 are set to zero.

[084] FIG. 4 illustrates logic implemented in the incremental virtual copy process 116 for performing an incremental virtual copy operation in accordance with certain implementations of the invention. Control begins at block 400 with the incremental virtual copy process 116 receiving an incremental virtual copy operation. An incremental

virtual copy operation may be issued by host 140. Although not shown in the flow of FIG. 4, prior to receipt of the incremental virtual copy operation, one or more blocks of data may have been updated in the source storage 120 and/or the target storage 130, by, for example, a user at host 140.

[085] In block 410, the incremental virtual copy process 116 updates indicators in the target copy structure 200 with indicators in the source and target change recording structures 210 and 220. In certain implementations of the invention, the source change recording structure 210 is merged with the target change recording structure 220 using an "OR" operation, and the result of the "OR" operation is "OR'd" to the target copy structure 200.

[086] In block 420, after the target copy structure 200 has been updated, the incremental virtual copy process 116 updates indicators in the source change recording structure 210 to indicate that the source blocks of data have not been updated since the last incremental virtual copy operation. In certain implementations of the invention, all of the indicators in the source change recording structure 210 are set to zero. In block 430, the incremental virtual copy process 116 updates the indicators in the target change recording structure 220 to indicate that the target blocks of data have not been updated since the last incremental virtual copy operation. In certain implementations of the invention, all of the indicators in the target change recording structure 220 are set to zero.

[087] FIG. 5 illustrates logic implemented in write process 114 for processing of a write operation in accordance with certain implementations of the invention. Control begins at block 500 with the write process 114 receiving a request to write a block of data. In block 520, the write process 114 determines whether the block of data is in an incremental virtual copy relationship. If so, processing continues to block 530, otherwise, processing continues to block 560. In block 530, the write process 114 determines whether the block of data is in target storage 130. If so, processing continues to block 540, otherwise, processing continues to block 550.

[088] In block 540, the indicator for the block of data in the target change recording structure 220 is updated to indicate that the target block of data has changed since the last incremental virtual copy operation. In certain implementations of the invention, the indicator in the target change recording structure 220 is set to one. In block 550, the indicator for the block of data in the source change recording structure 210 is updated to indicate that the source block of data has changed since the last incremental virtual copy operation. In certain implementations of the invention, the indicator in the source change recording structure 210 is set to one. In block 560, the write operation is performed by the write process 114.

[089] FIG. 6 illustrates logic implemented in the read process 112 for processing of a read

operation in accordance with certain implementations of the invention. Control begins at block 600 with receipt of a request to read a block of data. In block 620, the read process 112 determines whether the block of data is a target in an incremental virtual copy relationship. If so, processing continues to block 630, otherwise, processing continues to block 660. In block 630, the read process determines whether an indicator for the block of data is set in the target copy structure to indicate that data is to be read from source storage 120. If so, processing continues to block 640, otherwise, processing continues to block 650.

[090] In block 640, the read process 112 reads (i.e., stages) the block of data from source storage 120. In block 650, the read process 112 reads (i.e., stages) the block of data from target storage 130. In block 660, the read process 112 performs a normal read of the block of data.

[091] FIG. 7 illustrates a background copy process in accordance with certain implementations of the invention. Control begins at block 700 with a determination that it is time to copy a block of data. In block 710, it is determined whether the indicator in the target copy structure 200 for the block of data indicates that the block of data has not been copied. If so, processing continues to block 730, otherwise processing continues to block 720. In block 720, a next block of data may be processed or, if there are no other blocks of data to be processed, this logic terminates.

[092] In block 730, the block of data is read in accordance with the logic of FIG. 6. In block 740, the block of data is destaged to target storage 130. In block 750, an indicator in the target copy structure 200 is updated for the block of data to indicate that the block of data has been copied. In certain implementations of the invention, the indicator in the target copy structure 200 is set to zero.

[093] Thus, in certain implementations of the invention, the incremental virtual copy operation is achieved by monitoring writes (i.e., updates) and recording changes to tracks for volumes participating in an instant virtual copy relation. After an initial instant virtual copy operation, tracks that have been updated on either a source or a target volume can be copied from the source volume to the target volume, without copying the entire source volume.

[094] Enterprise Storage Server, FlashCopy, and 3990 are registered trademarks or common law marks of International Business Machines Corporation in the United States and/or other countries.

[095] A dditional Implementation Details

[096] The described techniques for incremental virtual copy may be implemented as a method, apparatus or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The term "article of manufacture" as used herein refers to code or logic im-

plemented in hardware logic (e.g., an integrated circuit chip, Programmable Gate Array (PGA), Application Specific Integrated Circuit (ASIC), etc.) or a computer readable medium, such as magnetic storage medium (e.g., hard disk drives, floppy disks,, tape, etc.), optical storage (CDROMs, optical disks, etc.), volatile and nonvolatile memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, DRAMs, SRAMs, firmware, programmable logic, etc.). Code in the computer readable medium is accessed and executed by a processor. The code in which preferred embodiments are implemented may further be accessible through a transmission media or from a file server over a network. In such cases, the article of manufacture in which the code is implemented may comprise a transmission media, such as a network transmission line, wireless transmission media, signals propagating through space, radio waves, infrared signals, etc. Thus, the "article of manufacture" may comprise the medium in which the code is embodied. Additionally, the "article of manufacture" may comprise a combination of hardware and software components in which the code is embodied, processed, and executed. Of course, those skilled in the art will recognize that many modifications may be made to this configuration without departing from the scope of the present invention, and that the article of manufacture may comprise any information bearing medium known in the art.

[097] The logic of FIGs. 3-7 describes specific operations occurring in a particular order. In alternative implementations, certain of the logic operations may be performed in a different order, modified or removed. Moreover, operations may be added to the above described logic and still conform to the described implementations. Further, operations described herein may occur sequentially or certain operations may be processed in parallel, or operations described as performed by a single process may be performed by distributed processes.

[098] The illustrated logic of FIGs. 3-7 may be implemented in software, hardware, programmable and non-programmable gate array logic or in some combination of hardware, software, or gate array logic.

[099] FIG. 8 illustrates an architecture of a computer system that may be used in accordance with certain implementations of the invention. A storage controller 100 and/or host 140 may implement computer architecture 800. The computer architecture 800 may implement a processor 802 (e.g., a microprocessor), a memory 804 (e.g., a volatile memory device), and storage 810 (e.g., a non-volatile storage area, such as magnetic disk drives, optical disk drives, a tape drive, etc.). An operating system 805 may execute in memory 804. The storage 810 may comprise an internal storage device or an attached or network accessible storage. Computer programs 806 in storage 810 may be loaded into the memory 804 and executed by the processor 802 in a manner known in the art. The architecture further includes a network card 808 to enable com-

munication with a network. An input device 812 is used to provide user input to the processor 802, and may include a keyboard, mouse, pen-stylus, microphone, touch sensitive display screen, or any other activation or input mechanism known in the art. An output device 814 is capable of rendering information transmitted from the processor 802, or other component, such as a display monitor, printer, storage, etc. The computer architecture 800 of the computer systems may include fewer components than illustrated, additional components not illustrated herein, or some combination of the components illustrated and additional components.

[100] The computer architecture 800 may comprise any computing device known in the art, such as a mainframe, server, personal computer, workstation, laptop, handheld computer, telephony device, network appliance, virtualization device, storage controller, etc. Any processor 802 and operating system 805 known in the art may be used.

[101] The foregoing description of implementations of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto. The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many implementations of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.

Claims

[001] A method for reducing an amount of data transferred, comprising maintaining a first indicator for each source block of data to indicate whether the source block of data has been updated in source storage since the source block of data was last transferred to target storage; maintaining a second indicator for each target block of data in the target storage to indicate whether the target block of data has been updated in the target storage since the target block of data was overwritten by a corresponding source block of data; and when transferring data from the source storage to the target storage, transferring each source block of data for which a first indicator has been set to indicate that the source block of data has been updated; and transferring each source block of data that corresponds to a target block of data for which a second indicator has been set to indicate that the target block of data has been updated.

[002] The method of claim 1, further comprising: maintaining a third indicator for each target block of data to indicate whether a corresponding source block of data or the target block of data is to be retrieved for a stage operation.

[003] The method of claim 2, further comprising: when a source block of data has been copied to the target storage, updating the third indicator for the target block of data.

[004] The method of claim 2 or claim 3, further comprising: receiving an incremental virtual copy operation; and updating the third indicator for each target block of data based on the first indicator for the source block of data and the second indicator for the target block of data corresponding to the source block of data, wherein transferring data from the source storage to the target storage comprises transferring each source block of data for which a third indicator indicates that the source block of data is to be transferred to the target storage.

[005] The method of any one of claims 2 to 4, further comprising: after establishing an incremental virtual copy relationship, updating the third indicator for each target block of data to indicate that the source block of data corresponding to the target block of data is to be retrieved from the source storage for a stage operation; after transferring data from the source storage to the target storage, updating the first indicator for each source block of data to indicate that the source block of data has not been updated in the source storage since the source block of data was last transferred to the target storage; and after transferring data from the source storage to the target storage, updating the second indicator for each target block of data to indicate that the target block of data has not been updated in the target storage since the source block of data was last transferred to the target storage.

[006] A system for reducing an amount of data transferred, comprising: means for maintaining a first indicator for each source block of data to indicate whether the source block of data has been updated in source storage since the source block of data was last transferred to target storage; means for maintaining a second indicator for each target block of data in the target storage to indicate whether the target block of data has been updated in the target storage since the target block of data was overwritten by a corresponding source block of data; and means for, when transferring data from the source storage to the target storage, transferring each source block of data for which a first indicator has been set to indicate that the source block of data has been updated; and means for transferring each source block of data that corresponds to a target block of data for which a second indicator has been set to indicate that the target block of data has been updated.

[007] The system of claim 6, further comprising: means for maintaining a third indicator for each target block of data to indicate whether a corresponding source block of data or the target block of data is to be retrieved for a stage operation.

[008] The system of claim 6 or claim 7, further comprising: means for, when a source block of data has been copied to the target storage, updating the third indicator for the target block of data.

[009] The system of any of claims 6, 7 or 8, further comprising: means for receiving an incremental virtual copy operation; means for updating the third indicator for each target block of data based on the first indicator for the source block of data and the second indicator for the target block of data corresponding to the source block of data, wherein transferring data from the source storage to the target storage comprises transferring each source block of data for which a third indicator indicates that the source block of data is to be transferred to the target storage; means for, after establishing an incremental virtual copy relationship, updating the third indicator for each target block of data to indicate that the source block of data corresponding to the target block of data is to be retrieved from the source storage for a stage operation; means for, after transferring data from the source storage to the target storage, updating the first indicator for each source block of data to indicate that the source block of data has not been updated in the source storage since the source block of data was last transferred to the target storage; and means for, after transferring data from the source storage to the target storage, updating the second indicator for each target block of data to indicate that the target block of data has not been updated in the between a source portion of data and a target portion of data by performing an incremental virtual copy operation between the source portion of data and the target portion of data, wherein the first indicator and the second indicator are

maintained for each source block of data and target block of data in the source and target portions of data in the incremental virtual copy relationship, respectively.

[010] A computer program comprising computer program code to, when loaded into a computer system and executed thereon, cause said compter system to perform the steps of a method according to any one of claims 1 to 5.