Abstract: A primary data storage system is connected with a separate and external active archive storage system to consolidate data and allow active archive data to be managed based on primary storage system events. The primary data storage system may be managed and maintained by an external entity, and may include a manager module such as a resource manager. The active archive system may include several tiers of storage in a hierarchical storage system and logic for moving data between and among the tiers. As data processing milestones are completed or the state of data changes, in projects stored in the primary data storage system, task milestone or state change events are detected. Event detection can trigger data movement in the active archive solution. One or more software modules implementing the present invention may detect the events and trigger active archive operations based on the events.
ACTIVE ARCHIVE BRIDGE

BACKGROUND

Companies are creating and processing more and more data, and need a reliable means to store and access the data in order to be competitive. As a result, the data storage industry is growing quickly. Data storage providers are using many solutions for storing the ever-increasing volume of data. Currently, companies use techniques such as compression and other new storage methods to reduce the overall footprint of the data. Despite these methods, the volume of data continues to strain the capacity of many existing data storage systems and there is still a need to improve data storage techniques.
SUMMARY

The present technology connects a primary data storage system with a separate and external active archive storage system to consolidate data and allow archived data to be managed based on primary storage system events. The primary data storage system may be managed and maintained by an external entity, and may include one or more manager modules such as a resource manager. The active archive system may include several tiers of storage in a hierarchical storage system and logic for moving data between and among the tiers. As data processing task milestones are completed or the state of data changes in projects stored in the primary data storage system, milestones or state change events are detected.

Data movement can be triggered by a condition that is active, such as a placement of new or altered data that triggers a policy for data placement in one of the tiers, or it could be triggered by something passive, such as for example when a file ages to the point that it now qualifies to be move based upon the fact that no one has touched it for a certain amount of time. Event detection can trigger data movement in the active archive solution. In embodiments, one or more software modules implementing the present invention may detect the events and trigger active archive operations based on the events. The one or more modules may be located on the primary data storage system, the active archive system, and distributed over both.

In an embodiment, a method for managing archive data may begin with detecting an event associated with a data file update in a first data system. A policy associated with the event may be accessed and data may be moved between tiers in an archive storage system based on the event and according to the policy. The first data system may be remote and external to the archive storage system.
BRIEF DESCRIPTION OF DRAWINGS

FIGURE 1 is a block diagram of a resource scheduling system.

FIGURE 2 is a block diagram of an active archive system.

FIGURE 3 is a method for archiving data based on resource scheduler events.

FIGURE 4 is a method for correlating project files with projects.

FIGURE 5 is a block diagram of a device for implementing the present technology.
DETAILED DESCRIPTION

The present technology connects a primary data storage system with a separate active archive storage system to consolidate data and allow active archive data to be based on primary storage system events.

The primary data storage system may be managed and maintained by an external entity. The primary storage system may include a resource manager module or other resource or workflow scheduler which provides access to storage within the system. Administrators of the external storage system may allow customers to store projects, each project having one or more files, on the primary storage system devices. Typically, project workflow may involve a first user completing a first task by moving data - which triggers an event - within the project which changes stored data, a second user completing a change in data which triggers another event within the project, and so on. As events occur, data associated with the completed project may be idle for long periods of time but are typically maintained in the primary storage system.

The active archive system may include several tiers of storage in a hierarchical storage system. The first tier may provide quick access to data at a higher power and costs, and may include solid state drives or primary hard disk drive storage. One or more lower tiers may include secondary hard disk drive storage, power-managed disk arrays (MAID) storage, tape storage, cloud storage, and other types of storage devices and systems which provide slower access times but at much cheaper costs. Active archive solutions may include logic for moving data within the data storage hierarchy - moving data vertically from tier to tier or horizontally among different storage devices within a tier. Hence, when data is received from an external source, the active archive solution includes logic for storing the data at an initial tier or tiers and for determining when the data should be moved to a different tier. However, typical active archive solutions are not compatible with primary data storage systems.

As data processing milestones are completed or the state of data changes, for example data files remaining un-accessed and un-altered for some pre-determined
period of time or other policy triggers, in projects stored in the primary data storage system, the milestone or state change events are detected. Event detection triggers data actions in the active archive solution. In embodiments, one or more software modules implementing the present invention may detect the events and trigger active archive operations based on the events. The one or more modules may be located on the primary data storage system, the active archive system, and distributed over both. The active archive operations performed in response to receipt of a trigger event may include initially storing data received from an external source, such as from primary storage, to an initial archive storage device or devices on one or more archive tiers, and applying automated policies or processing later events received by the active archive solution to move data vertically from tier to tier or horizontally among different storage devices within a tier.

FIGURE 1 is a block diagram of a resource scheduling system. The resource scheduling system of FIGURE 1 may operate as a primary data storage system and may include resource scheduler 110, network available storage (NAS) 120, 130, and 140, and storage area network (SAN) 150. Other hosts may include any device or system that provides a block-based storage system, file-based storage system, or object-based storage system.

Resource scheduler 110 may allow users of the primary data storage system to access and manipulate data stored in resources NAS 120-140 and SAN 150. A user may access and manipulate the data through a client device, such as client device 130. Users may access and change data files associated with a project. As a user changes a file of a project, the user completes a task and triggers an event and provides an indication that the task - the update or change to the file - is complete. After the user completes the task, another user may perform another task on a file within the project - which triggers another event. When a task is complete, the files that were updated, and sometimes the entire project, may be idle and left untouched for a long period of time.

Active archive system 170 may include several tiers of storage in a hierarchical storage system. A first tier may provide quick access to data at a higher power and costs and lower tiers may include MAID storage and tape storage which
provide slower access times but at much cheaper costs. Active archive 170 may include logic for managing and moving data between tiers based on data retention policies but inherently does not have any knowledge regarding the status of data within the primary data storage system.

Meta-data policy engine (MDPE) 160 may communicate with resource scheduler 110, NAS-120-140, SAN 150, and active archive 170. MDPE may include one or more modules that are able to scan the primary data storage system to collect project information, retrieve event information from resource scheduler 110, and trigger data movements within active archive 170 based on events associated with completed tasks. The MDPE may be located on a device or virtual machine between the primary data storage system and active archive 170, as shown by MDPE 160. The MDPE may also be implemented in the resource scheduler 110, active archive 170, or both. For example, an MDPE module 114 of the implemented in the resource scheduler may scan for events and communicate the events to an MDPE module 172 within active archive 170.

Client device 130, resource scheduler 110, NAS-120-140, SAN 150 and active 170 may communicate via one or more private networks, public networks, WANs, LANs, an intranet, the Internet, or a combination of these networks.

FIGURE 2 is a block diagram of an active archive system. The active archive system of FIGURE 2 provides more detail for active archive 170 of the system of FIGURE 1. The active archive system of FIGURE 2 includes gateway 210, disc storage 220, maid 230, tape library 240 and cloud based archive 250. In embodiments, the gateway 210 may include all or a subset of downstream storage devices and systems 210-250, implemented within a single appliance or machine. Gateway 210 may include an instance of MDPE 114 and may communicate with devices configured to provide data to the active archive system. Gateway 210 may include all or a portion of the logic for moving data from one tier to another within the active archive system according to a set of retention policies. For example, the retention policies may specify that data may be moved from tier one storage to tier two or tier three storage based on how old the file is, the file name, a group ID for the file, and so on.
Disc storage 220 may form a high tier or first tier within the tiered active archive storage system. Disc storage 220 may provide fast access to data but at a higher price as compared to other storage tiers. MAID 230 may form a middle tier of the active archive system having slower access than disc storage 220 at a lower price. Tape library 240 is a lowest tier of the active archive system. Tape library 240 may store data very cheaply but with the slowest access time. Cloud based archive 250 may be used to expand the storage space of a tier 1 or tier 2 of the active archive system.

MDPE 114 may communicate with the resource scheduler 110. MDPE 114 may, for example, retrieve project file information, scan resource scheduler 110 (or other locations) for events, and may implement policy-based and other data movements within active archive storage.

FIGURE 3 is a method for archiving data based on resource scheduler events. The method of FIGURE 3 may be performed by MDPE modules on primary data storage system, active archive 170, in between the two systems, or distributed over multiple locations. The method of FIGURE 3 begins with identifying project files within projects at step 310. The project file architecture is required to move data within active archive 170 when events associated with those projects have been detected. More data regarding identifying project files within projects is discussed below with respect to FIGURE 4.

Next, a resource scheduler external to the active archive 170 is scanned for events at step 320. If an event is not detected at step 330, the process of FIGURE 3 returns to step 320. If an event such as a completed task event is detected within the resource scheduler, a policy associated with the event may be accessed at step 340. The policy may be stored at active archive 170, at a device implementing MDPE 160, or elsewhere. The policy may indicate how data associated with a project is to be handled within active archive 170 when a corresponding event is detected. For example, the policy may be to move a project from one location in the primary data storage infrastructure to another location within the storage infrastructure. Additionally, the policy may specify to move the data from a high tier storage type to a lower tier storage type. In any event, once an event is detected and the policy
associated with the event is accessed, data is moved in the active archive according to the access policy at step 350. In embodiments, the data may be moved, copied and/or selected with the option of administrator confirmation before the move occurs at step 350.

FIGURE 4 is a method for correlating project files with projects. The method of FIGURE 4 provides more detail for step 310 of the method of FIGURE 3. First, the existing file system of the resource scheduler is scanned at step 410. The existing file system may include the primary data storage infrastructure and files stored on the system resources, such as NAS 120-140 and SAN 150. The file system details at step 420. A scheduler may then be queried for project identifiers and project files at step 430. The project files in the file system may then be correlated with the projects at step 440. In embodiments, the file system details include file location information. The location of each file is associated with a corresponding project identifier. The location may be associated with the project, such as if it is in a project folder. But there may be occasions when a project file is also in a large collection of other files in a disorganized folder. In that case, the MDPE or the resource scheduler could be used to link files to the project. When an event is detected with a particular project, files at the locations associated with the project may be handled together in a data move from one project to another.

FIGURE 5 illustrates an exemplary computing system 500 that may be used to implement a computing device for use with the present technology. System 500 of FIGURE 5 may be implemented in the contexts of the likes of client devices and systems 105, 110, 120-150, 160, and 210. The computing system 500 of FIGURE 5 includes one or more processors 510 and memory 520. Main memory 520 stores, in part, instructions and data for execution by processor 510. Main memory 520 can store the executable code when in operation. The system 500 of FIGURE 5 further includes a mass storage device 530, portable storage medium drive(s) 540, output devices 550, user input devices 560, a graphics display 570, and peripheral devices 580.

The components shown in FIGURE 5 are depicted as being connected via a single bus 590. However, the components may be connected through one or more
data transport means. For example, processor unit 510 and main memory 520 may be connected via a local microprocessor bus, and the mass storage device 530, peripheral device(s) 580, portable storage device 540, and display system 570 may be connected via one or more input/output (I/O) buses.

Mass storage device 530, which may be implemented with a magnetic disk drive or an optical disk drive, is a non-volatile storage device for storing data and instructions for use by processor unit 510. Mass storage device 530 can store the system software for implementing embodiments of the present invention for purposes of loading that software into main memory 520.

Portable storage device 540 operates in conjunction with a portable non-volatile storage medium, such as a floppy disk, compact disk or Digital video disc, to input and output data and code to and from the computer system 500 of FIGURE 5. The system software for implementing embodiments of the present invention may be stored on such a portable medium and input to the computer system 500 via the portable storage device 540.

Input devices 560 provide a portion of a user interface. Input devices 560 may include an alpha-numeric keypad, such as a keyboard, for inputting alpha-numeric and other information, or a pointing device, such as a mouse, a trackball, stylus, or cursor direction keys. Additionally, the system 500 as shown in FIGURE 5 includes output devices 550. Examples of suitable output devices include speakers, printers, network interfaces, and monitors.

Display system 570 may include a liquid crystal display (LCD) or other suitable display device. Display system 570 receives textual and graphical information, and processes the information for output to the display device.

Peripherals 580 may include any type of computer support device to add additional functionality to the computer system. For example, peripheral device(s) 580 may include a modem or a router.

The components contained in the computer system 500 of FIGURE 5 are those typically found in computer systems that may be suitable for use with embodiments of the present invention and are intended to represent a broad category of such computer components that are well known in the art. Thus, the computer system 500
of FIGURE 5 can be a personal computer, hand held computing device, telephone, mobile computing device, workstation, server, minicomputer, mainframe computer, or any other computing device. The computer can also include different bus configurations, networked platforms, multi-processor platforms, etc. Various operating systems can be used including Unix, Linux, Windows, Macintosh OS, Palm OS, and other suitable operating systems.

The foregoing detailed description of the technology herein has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the technology to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. The described embodiments were chosen in order to best explain the principles of the technology and its practical application to thereby enable others skilled in the art to best utilize the technology in various embodiments and with various modifications as are suited to the particular use contemplated. It is intended that the scope of the technology be defined by the claims appended hereto.
CLAIMS

What is claimed is:

1. A method for managing archive data, comprising:
   detecting an event associated with a data file in a first data system;
   accessing policy associated with the event; and
   moving data between two or more tiers in an archive storage system based on
   the event and according to the policy, the first data system remote and external from
   the archive storage system.

2. The method of claim 1, wherein the event is associated with a completed task or a
   passive event.

3. The method of claim 1, wherein the data includes a plurality of files associated
   with a project.

4. The method of claim 3, wherein the event is associated with the project.

5. The method of claim 1, wherein the event is detected by scanning a resource
   scheduler which implements the data file update.

6. The method of claim 1, wherein the policy as accessed by a module stored and
   executed within the archive storage system.

7. The method of claim 1, further comprising scanning the first data system to
   identify files associate with a project, the moved data associated with the project.
8. A computer readable non-transitory storage medium having embodied thereon a program, the program being executable by a processor to perform a method for managing archive data, the method comprising:
   detecting an event associated with a data file in a first data system;
   accessing policy associated with the event; and
   moving data between two or more tiers in an archive storage system based on the event and according to the policy, the first data system remote and external from the archive storage system.

9. The computer readable non-transitory storage medium of claim 8, wherein the event is associated with a completed task.

10. The computer readable non-transitory storage medium of claim 8, wherein the data includes a plurality of files associated with a project.

11. The computer readable non-transitory storage medium of claim 10, wherein the event is associated with the project.

12. The computer readable non-transitory storage medium of claim 8, wherein the event is detected by scanning a resource scheduler which implements the data file update.

13. The computer readable non-transitory storage medium of claim 8, wherein the policy as accessed by a module stored and executed within the archive storage system.

14. The computer readable non-transitory storage medium of claim 8, the method further comprising scanning the first data system to identify files associate with a project, the moved data associated with the project.
15. A system for managing archive data, the system comprising:

- a processor;
- a memory;
- one or more modules stored in the memory and executed by the processor to:
  - detect an event associated with a data file in a first data system;
  - access policy associated with the event; and
  - move data between tiers in an archive storage system based on the event and according to the policy, the first data system remote and external from the archive storage system.

16. The system of claim 1, wherein the event is associated with a completed task.

17. The system of claim 1, wherein the data includes a plurality of files associated with a project.

18. The system of claim 3, wherein the event is associated with the project.

19. The system of claim 1, wherein the event is detected by scanning a resource scheduler which implements the data file update.

20. The system of claim 1, wherein the policy as accessed by a module stored and executed within the archive storage system.

21. The system of claim 1, the one or more modules further executable to scan the first data system to identify files associated with a project, the moved data associated with the project.
FIGURE 2
Identify project files within projects

Scan external resource manager for events

Event detected?

Access policy associated with event

Move data in active archive according to accessed policy

FIGURE 3
Scan existing file system

Store file system details

Query scheduler for project identifiers and project files

Correlate project files in file system with projects

FIGURE 4
INTERNATIONAL SEARCH REPORT

International application No. PCT/US 14/17291

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - G06F 12/00 (2014.01)
USPC - 7 11/165

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
USPC: 7 11/165

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 711/100, 154, 161, 162, 165 (keyword limited - see search terms below)

Electronic database consulted during the international search (name of database and, where practicable, search terms used)
PatBase; Google; Google Scholar
Terms: storage, memory, archive, layer, level, tier, event, instance, file, data, policy, rule, remote, trigger, package.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category*</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>US 7,617,261 B2 (Telkowski et al.) 10 November 2009 (10.1.2009), entire document, especially abstract. Figs. 1, 11, col. 3, l. 8-14, col. 3, l. 25-44, col. 4, l. 5-16, col. 4, l. 27-44, col. 4, l. 58 to col. 5, l. 10, col. 10, l. 13-24, col. 10, l. 61 to col. 11, l. 3, col. 12, l. 4-12, col. 13, l. 55-67, col. 14, l. 35-43, col. 14, l. 52-64, col. 15, l. 11-20, claim 1, claim 3.</td>
<td>1-21</td>
</tr>
<tr>
<td>A</td>
<td>US 7,693,891 B2 (Stokes et al.) 06 April 2010 (06.04.2010), entire document.</td>
<td>1-21</td>
</tr>
<tr>
<td>A</td>
<td>US 2010/014591 7 A1 (Bone et al.) 10 June 2010 (10.06.2010), entire document.</td>
<td>1-21</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C.

* Special categories of cited documents:
  "A" document defining the general state of the art which is not considered to be of particular relevance
  "E" earlier application or patent but published on or after the international filing date
  "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  "O" document referring to an oral disclosure, use, exhibition or other means
  "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 10 June 2014 (10.06.2014)

Date of mailing of the international search report 17 JUL 2014

Name and mailing address of the ISA/US
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450
Facsimile No. 571-273-3201

Authorized officer: Lee W. Young
PCT Helpdesk: 571-272-4300
PCT OSP: 571-272-7774