

METHOD AND SYSTEM FOR CREATING AND MAINTAINING VERSION-SPECIFIC PROPERTIES IN A DISTRIBUTED ENVIRONMENT

TECHNICAL FIELD

[0001] This invention relates generally to distributed computing environments and more particularly to the format and management of object properties stored as part of each object in a particular system. More particularly still, the present invention relates to the use of object properties to provide status information to third party applications.

BACKGROUND OF THE INVENTION

[0002] In a distributed environment, such as the Internet, computer information is typically stored in units known as “resources” or “objects.” Resources may be any entity that can be accessed on the web through a uniform resource locator (URL) or uniform resource identifier (URI). Groups of member resources may be collected into what is known as a “collection.” One particular benefit to grouping resources relates to the fact that once a group of resources are collected, they may be treated in a similar manner as computer files in a non-distributed environment, wherein both the files and the resource collections each have data and metadata. The data is the actual object data for either the object or the resource and the metadata is the information that describes the file or resource, and possibly even the data.

[0003] As the Internet and other distributed environments increase in popularity, advances are being made in the representative functionality provided to client systems as the client systems interact with server systems and the resources on the server systems. For instance, past protocols, such as Hypertext Transfer Protocol (HTTP) version 1.1 provided a client the ability to “put” information into a server-side application or document. The client used a particular Hypertext Markup Language (HTML) request that included a “PUT” command in the header. The server then interpreted this header information and executed the command.

[0004] Recent developments have increased the concept of authoring resources on a server system from client system over the distributed network. One particular example relates to the development of the WebDAV standard, which stands for the World Wide Web Distributed Authoring and Versioning standard, referred to herein as simply “DAV.” DAV provides a set of headers and methods, which extend the HTTP to provide capabilities for overwrite prevention (locking), properties, and namespace management. A document that was written by the IETF and has been approved by the IESG as Proposed Standard RFC 2518, published February 1999, the standard describes DAV in more detail.

[0005] DAV provides for the creation of property objects that are associated with a resource, wherein the properties are similar to properties of a file in a more traditional file system. The properties may store information such as the name of the resource, the author of the resource, the latest version value for the resource (or collection of resources), among others.

[0006] Often, third party applications work in combination with a server-side resource systems to provide additional system features or functions, such as virus scanning functions. These third party applications may actually “inter-

cept” each resource access attempt and scan the object for viruses or perform other tests prior to performing the actual access operation. Unfortunately however, performing a scan operation or other tests each time a resource is accessed consumes a significant amount of time. Therefore, a log of information may be maintained to store version information for each resource. For example, the log may maintain a list of resources on the system and whether each resource has been scanned, and if so, which version of virus definition file was used. Using the log of information, the virus scanner can reduce processing time by only scanning resources that are new or modified, or that were scanned by an out-of-date virus definition file.

[0007] Although the log may provide a significant improvement over systems that scan every resource before each access, using such a log suffers from some drawbacks. For example, maintaining such an information log requires a significant amount of overhead. Moreover, the process of accessing the log to determine whether items have been scanned reduces overall system performance since a separate file must be located on disk and examined. Additionally the log of information is not updated as resources are copied or backed-up causing unnecessary scanning operations in certain circumstances.

[0008] One solution to the performance problems associated with keeping an information log as a separate file has been to keep an “in-memory” log that is created and stored in volatile memory, e.g., RAM. The in-memory log may be accessed more quickly than a separate file, and therefore performance of the system increases when using the in-memory log in place of the log file described above. However, the in-memory log is erased or lost when the power is not delivered to the system, such as when the system is turned off, shut down or rebooted. Thus, any state information or version information may not be determined quickly following a cessation of power. Another drawback associated with the in-memory log is that such a log consumes a significant amount of the operating memory used by the system. Therefore, the in-memory log has not provided an adequate solution to the above-described problems.

[0009] It is with respect to these and other considerations that the present invention has been made.

SUMMARY OF THE INVENTION

[0010] The present invention solves these problems by creating and using a version-specific attribute or property that is stored as part of a resource or object collection, contains specific version information relating to how or when the property was created and is automatically invalidated when a predetermined “update” event occurs. Third party applications may generally create and access these versions-specific properties to thereby eliminate the need for external logs or databases.

[0011] An aspect of the present invention relates to a method of providing version-specific information associated with an object stored in a computer system to an application. The method involves creating a version-specific property that is associated with the object, maintaining the version-specific property to reflect relevant updates to the object by automatically invalidating the version-specific information in response to predetermined access requests and allowing the third party application to analyze the version-specific property.