SYNCHRONIZING FILE PARTITIONS UTILIZING A SERVER STORAGE MODEL

BACKGROUND

[0001] In client-server computer networks, an electronic document file may be edited by multiple users at different client computers. The electronic document file may include associated editing session data (e.g., metadata) which is different and independent from the main document file content. The electronic document file is synchronized between a client computer and a central server in order to store any edits made to the electronic document file which may then be viewed by users of other client computers in a network. Under existing synchronization methods, any associated metadata must be synchronized together with the edits made to the electronic document file. In particular, for markup language or object linking and embedding ("OLE") compound documents, metadata may be put in extensible file format for synchronization. As a result of the synchronization, the metadata associated with an electronic document file is automatically exposed on the server, even if a user synchronizing the electronic document file does not desire the metadata to be viewed by or downloaded (within the electronic document file) to other users. It is with respect to these considerations and others that the various embodiments of the present invention have been made.

SUMMARY

[0002] This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended as an aid in determining the scope of the claimed subject matter.

[0003] Embodiments are provided for synchronizing file partitions utilizing a server storage model in a client-server computer network. Edits to content of an electronic document may be received at a client computer. The content may be included in a first partition of a file stored on a server computer. The first partition may include a first stream which includes the electronic document content. The file may include multiple partitions with each partition including one or more streams. The client computer may generate metadata in a second partition of the file. The metadata may be associated with the edits to the electronic document content in the first partition. The second partition may then be individually synchronized with the server computer to store the metadata. The

first partition may then be individually synchronized with the server computer to store the edits made to the electronic document content.

[0004] These and other features and advantages will be apparent from a reading of the following detailed description and a review of the associated drawings. It is to be understood that both the foregoing general description and the following detailed description are illustrative only and are not restrictive of the invention as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] FIGURE 1 is a block diagram illustrating a client-server network architecture which may utilize a server storage model for synchronizing file partitions, in accordance with various embodiments;

[0006] FIGURE 2 is a block diagram illustrating a client computing environment which may utilize a server storage model for synchronizing file partitions, in accordance with various embodiments;

[0007] FIGURE 3 is a flow diagram illustrating a routine for synchronizing file partitions utilizing a server storage model, in accordance with an embodiment;

[0008] FIGURE 4 is a flow diagram illustrating a routine for synchronizing a single file partition utilizing a server storage model, in accordance with the embodiment illustrated in FIGURE 3;

[0009] FIGURE 5 is a flow diagram illustrating a routine for synchronizing another file partition utilizing a server storage model, in accordance with the embodiment illustrated in FIGURE 3;

[0010] FIGURE 6 is a flow diagram illustrating a routine for synchronizing file partitions utilizing a server storage model, in accordance with another embodiment; and

[0011] FIGURE 7 is a flow diagram illustrating a routine for synchronizing a file partition on a second client computer utilizing a server storage model, in accordance with the embodiment illustrated in FIGURE 6.

DETAILED DESCRIPTION

[0012] Embodiments are provided for synchronizing file partitions utilizing a server storage model in a client-server computer network. Edits to content of an electronic document may be received at a client computer. The content may be included in a first partition of a file stored on a server computer. The first partition may include a first stream which includes the electronic document content. The file may include multiple partitions with each partition including one or more streams. The client computer may generate metadata in a second partition of the file. The metadata may be associated with

the edits to the electronic document content in the first partition. The second partition may then be individually synchronized with the server computer to store the metadata. The first partition may then be individually synchronized with the server computer to store the edits made to the electronic document content.

[0013] In the following detailed description, references are made to the accompanying drawings that form a part hereof, and in which are shown by way of illustrations specific embodiments or examples. These embodiments may be combined, other embodiments may be utilized, and structural changes may be made without departing from the spirit or scope of the present invention. The following detailed description is therefore not to be taken in a limiting sense, and the scope of the present invention is defined by the appended claims and their equivalents.

[0014] Referring now to the drawings, in which like numerals represent like elements through the several figures, various aspects of the present invention will be described. FIGURE 1 is a block diagram illustrating a client-server network architecture which may utilize a server storage model for synchronizing file partitions, in accordance with various embodiments. The network architecture includes a client computer 2 and a client computer 6, both of which are in communication with a server computer 70 over a network 4. The network 4 which may include a local network or a wide area network (e.g., the Internet).

[0015] The client computer 2 may store a productivity application 30 which may be used to edit an electronic document file 40 (also stored on the client computer 2). In accordance with various embodiments, the productivity application 30 may comprise the WORD word processing software, the POWERPOINT presentation graphics program, and the GROOVE software from MICROSOFT CORPORATION of Redmond, Washington. It should be appreciated, however, that other productivity or application programs from other manufacturers may be utilized in accordance with the various embodiments described herein.

[0016] The electronic document file 40 may be received at the client computer 2 via a download operation over the network 4 from the server computer 70 or generated by the productivity application 30 on the client computer 30. The electronic document file 40 may include a partition 50. The partition 50 may include streams 52 and 56. The stream 52 may further include content (i.e., electronic document content) 54A. The electronic document file 40 may also include a partition 60. The partition 60 may include streams 62 and 68. The stream 62 may further include metadata 64A. In accordance with an

embodiment, the metadata 64A may be associated with the content 54A stored in the partition 50. In particular, the metadata 64A may include an author list 67A describing one or more authors of the content 54A and document locks 66A describing a current user of the electronic document file 40 (e.g., a user currently editing the content 54A).

[0017] In accordance with various embodiments, a "partition" is defined as a "file" within a file (such as the electronic document file 40) and may contain one or more streams. A "stream," which is contained within a partition, contains data associated with an electronic document file (such as the content 54A or the metadata 64A). It should be understood that each of the partitions in a file may be synchronized independently of each other and at different synchronization rates (i.e., partitions in a single file may be synchronized at different times). Furthermore, multiple streams in the same partition are related to each other while no relationship may exist between streams in different partitions. For example, stream 56 comprise data associated with electronic document content (and thus related to stream 52) but may not contain metadata associated with the electronic document file 40. In addition, multiple streams in a given partition are synchronized simultaneously, thereby ensuring consistency between streams in the same partition.

[0018] The client computer 6 may store the productivity application 30 which also may be used to edit the electronic document file 40 (also stored on the client computer 6). The electronic document file 40 may be received at the client computer 6 via a download operation over the network 4 from the server computer 70 or generated by the productivity application 30 on the client computer 6. The electronic document file 40 may include the partition 50. The partition 50 may include the streams 52 and 56. The stream 52 may further include content (i.e., electronic document content) 54B. The electronic document file 40 may also include the partition 60. The partition 60 may include the streams 62 and 68. The stream 62 may further include metadata 64B. It should be understood, that in accordance with various embodiments, "metadata" represents content contained with a stream in a partition. In accordance with an embodiment, the metadata 64B may be associated with the content 54B stored in the partition 50. In particular, the metadata 64B may include an author list 67B describing one or more authors of the content 54B and document locks 66B describing a current user of the electronic document file 40 (e.g., a user currently editing the content 54B).

[0019] The server computer 70 may store a server application 42. In accordance with various embodiments, the server computer 70 may function as a document storage system for electronic files created on the client computers 2 and 6. The server application 42 may

comprise a collaborative services application program such as the SHAREPOINT SERVER services application from MICROSOFT CORPORATION. As is known to those skilled in the art, SHAREPOINT services technology enables users to create, maintain, and present a collaborative environment to share information. Using the technology, a user or organization can create one or more websites to provide and share information (e.g., documents on a web server or web folder, etc.) for other users associated with the websites. It should be understood that the embodiments described herein should not be construed as being limited to SHAREPOINT services technology and that other collaborative services technology from other developers and/or manufacturers may also be utilized.

[0020] The server computer 70 may also store the electronic document file 40. The electronic document file 40 may include the partition 50. The partition 50 may include the streams 52 and 56. The stream 52 may further include content (i.e., electronic document content) 54. The electronic document file 40 may also include the partition 60. The partition 60 may include the streams 62 and 68. The stream 62 may further include metadata 64. In accordance with an embodiment, the metadata 64 may be associated with the content 54 stored in the partition 50. In particular, the metadata 64 may include an author list 67 describing one or more authors of the content 54 and document locks 66 describing a current user of the electronic document file 40 (e.g., a user currently editing the content 54). It should be understood, that in accordance an embodiment, the electronic document file 40 may be downloaded from the server computer 70 to the client computers 2 and 6 for editing thereby potentially creating different versions of the content 54 and the metadata 64 in the electronic document file 40 when changes are made prior to being synchronized back to the server computer 70. Thus, in the present detailed description, versions of the content 54 and the metadata 64 on the client computers 2 and 6 are identified by the letters "A" and "B" appended after the reference numerals for these file components.

[0021] In accordance with various embodiments and as will be described in greater detail below with respect to FIGURES 3-7, the server application 42 and the productivity application 30 may be configured to facilitate the synchronization of the partitions 50 and 60 on the client computers 2 and 6, with the server computer 70. In accordance with an embodiment, the server application 42 may be configured to utilize a protocol (including, but not limited to a stateless protocol) to facilitate incremental document synchronization and to add support for server files to expose multiple independent or dependent data

streams. Thus, each of the partitions in an electronic file may be synchronized independently of each other and at different synchronization rates (i.e., partitions in a single file may be synchronized at different times). For example, the server application 42 in conjunction with the productivity application 30, may be configured to synchronize the partition 60 (containing the metadata 64) on the client computer 2 with the server computer 70 before the partition 50 (containing the content 54A). It should be appreciated that by synchronizing partitions at different rates, the amount of data communicated to the server computer 70 over the network 4 is reduced, thereby improving server scalability. An illustrative stateless protocol which may be utilized in accordance with various embodiments is described in related patent application U.S. Serial No. 12/113,975 entitled "Document Synchronization over Stateless Protocols," filed on May 2, 2008, the disclosure of which is incorporated herein, in its entirety, by reference.

[0022] It should be understood that that the various partitions and streams discussed with respect to the client computers 2 and 6, and the server computer 70, are merely illustrative and not limited to the number of partitions or streams which may be contained within an electronic document file in accordance with various embodiments. For example, in accordance with an embodiment, an electronic document file may contain only a single partition and a single stream while in accordance with other embodiments an electronic document file may contain multiple partitions and multiple streams. It should further be understood that, in accordance with various embodiments, the content of the streams contained within an electronic document file on the client computers 2 and 6 and the server computer 70 are not limited to metadata but may also include generic data (i.e., data which is not associated with content stored in an electronic document file partition) as well. It should further be understood that the embodiments described herein should not be construed as being limited to the aforementioned software applications and that other software applications from other developers and/or manufacturers may also be utilized. It should further be understood that the network architecture of FIGURE 1 and its components include functionality to communicate with other computing devices, communication devices, and/or other systems and are not intended to be limited to the embodiments and examples described herein. Thus, for example, the network architecture of FIGURE 1 may include additional client computers in communication with the server computer 70, in accordance with various embodiments.

Exemplary Operating Environment

[0023] Referring now to FIGURE 2, the following discussion is intended to provide a brief, general description of a suitable computing environment in which various illustrative embodiments may be implemented. While various embodiments will be described in the general context of program modules that execute in conjunction with program modules that run on an operating system on a personal computer, those skilled in the art will recognize that the various embodiments may also be implemented in combination with other types of computer systems and program modules.

[0024] Generally, program modules include routines, programs, components, data structures, and other types of structures that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the various embodiments may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like. The various embodiments may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

[0025] FIGURE 2 shows the client computer 2 which may include a general purpose desktop, laptop, handheld, tablet, or other type of computer capable of executing one or more application programs. The client computer 2 includes at least one central processing unit 8 ("CPU"), a system memory 12, including a random access memory 18 ("RAM") and a read-only memory ("ROM") 20, and a system bus 10 that couples the memory to the CPU 8. A basic input/output system containing the basic routines that help to transfer information between elements within the computer, such as during startup, is stored in the ROM 20. The client computer 2 further includes a mass storage device 14 for storing an operating system 32, the productivity application 30, and the electronic document file 40 which includes the partition 50, the streams 52, 56, 62, and 68, the content 54A, the metadata 64A, the author list 67A, and the document locks 66A, described above with respect to FIGURE 1.

[0026] In accordance with various embodiments, the operating system 32 may be suitable for controlling the operation of a networked personal computer, such as the WINDOWS operating systems from MICROSOFT CORPORATION of Redmond, Washington. The mass storage device 14 is connected to the CPU 8 through a mass storage controller (not

shown) connected to the bus 10. The mass storage device 14 and its associated computer-readable media provide non-volatile storage for the client computer 2. Although the description of computer-readable media contained herein refers to a mass storage device, such as a hard disk or CD-ROM drive, it should be appreciated by those skilled in the art that computer-readable media can be any available media that can be accessed or utilized by the client computer 2. By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media.

[0027] Computer storage media includes volatile and non-volatile, removable and non-removable hardware storage media implemented in any physical method or technology for the storage of information such as computer-readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EPROM, EEPROM, flash memory or other solid state memory technology, CD-ROM, digital versatile disks ("DVD"), or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, which can be used to store the desired information and which can be accessed by the client computer 2. Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media. Combinations of any of the above should also be included within the scope of computer-readable media. Computer-readable media may also be referred to as a computer program product.

[0028] According to various embodiments, the client computer 2 may operate in a networked environment using logical connections to remote computers through the network 4 which may comprise, for example, a local network or a wide area network (e.g., the Internet). The client computer 2 may connect to the network 4 through a network interface unit 16 connected to the bus 10. It should be appreciated that the network interface unit 16 may also be utilized to connect to other types of networks and remote computing systems. The client computer 2 may also include an input/output controller 22 for receiving and processing input from a number of input types, including a keyboard, mouse, pen, stylus, finger, and/or other means. Similarly, an input/output controller 22

may provide output to a display device 82, a printer, or other type of output device. Additionally, a touch screen can serve as an input and an output mechanism. It should be appreciated that the client computer 6 and the server computer 70 shown in FIGURE 1 may include many of the conventional components shown with respect to the client computer 2 in FIGURE 2.

[0029] FIGURE 3 is a flow diagram illustrating a routine 300 for synchronizing file partitions utilizing a server storage model, in accordance with an embodiment. When reading the discussion of the routines presented herein, it should be appreciated that the logical operations of various embodiments of the present invention are implemented (1) as a sequence of computer implemented acts or program modules running on a computing system and/or (2) as interconnected machine logical circuits or circuit modules within the computing system. The implementation is a matter of choice dependent on the performance requirements of the computing system implementing the invention. Accordingly, the logical operations illustrated in FIGURES 3-7 and making up the various embodiments described herein are referred to variously as operations, structural devices, acts or modules. It will be recognized by one skilled in the art that these operations, structural devices, acts and modules may be implemented in software, in firmware, in special purpose digital logical, and any combination thereof without deviating from the spirit and scope of the present invention as recited within the claims set forth herein.

[0030] The routine 300 begins at operation 305, where the productivity application 30 executing on the client computer 2 receives edits to electronic document content (contained in the stream 52) in the partition 50 of the electronic document file 40. In accordance with an embodiment, the electronic document file 40 may be downloaded from the server computer 70 including all of the partitions and streams contained therein. The partition 50 may then be opened in order to edit the content 54 contained therein, thereby creating the content 54A.

[0031] From operation 305, the routine 300 continues to operation 310 where the productivity application 30 generates the metadata 64A in the partition 60 of the electronic document file 40. In particular, the productivity application 30 may generate metadata to update the author list 67 in the partition 60 of the electronic document file 40 to add the user of the client computer 2 as an author (thereby generating the author list 67A). The productivity application 30 may also generate metadata to add a document lock 66A to the partition 60 of the electronic document file 40 to indicate that the electronic document file 40 is currently being edited by the user of the client computer 2.

[0032] From operation 310, the routine 300 continues to operation 315 where the productivity application 30 sends a request to the server application 42 to synchronize the partition 60 with the server computer 70 in order to save the changes made to the metadata 64 (i.e., the metadata 64A containing the author list 67A and the document lock 66A) to the server computer 70. It should be understood that the partition 60 may be synchronized independently from the partition 50 such that only the changed metadata 64A is saved to the server computer 70. An illustrative routine for synchronizing the partition 60 with the server computer will be described in greater detail below with respect to FIGURE 4.

[0033] From operation 315, the routine 300 continues to operation 320 where the productivity application 30 sends a request to the server application 42 to synchronize the partition 50 with the server computer 70 in order to save the edited content 54A to the server computer 70. It should be understood that the partition 50 may be synchronized independently from the partition 60 such that only the edited content 54A is saved to the server computer 70. It should further be understood that the order of synchronization of the partitions 60 and 50 may be reversed such that the partition 50 is synchronized independently with the server computer 70 prior to the partition 60. An illustrative routine for synchronizing the partition 50 with the server computer 70 will be described in greater detail below with respect to FIGURE 5. From operation 320, the routine 300 continues to operation 325 where it then ends. It should be understood that the operations 310-320, discussed above, may also be performed by the productivity application 30 executing on the client computer 6, in accordance with an embodiment.

[0034] FIGURE 4 is a flow diagram illustrating a routine 400 for synchronizing a single file partition utilizing a server storage model, in accordance with the embodiment illustrated in FIGURE 3. The routine 400 begins from operation 315 of FIGURE 3 and continues to operation 405, where the server application 42 determines whether a conflict exists with synchronizing the partition 60 on the client computer 2 with the server computer 70. In particular and in accordance with an embodiment, the server application 42 may compare the metadata 64A stored in the partition 60 on the client computer 2 with the metadata 64 stored on the server computer 70 to determine any differences. If the metadata 64A and the metadata 64 are different, then the server application 42 determines there is a conflict and the routine 400 continues to operation 410. If there are no differences between the metadata 64A and the metadata 64, then the routine 400 returns to operation 320 of FIGURE 3.

[0035] At operation 410, the server application 42 determines how to resolve the conflict based on whether the metadata 64A in the partition 60 on the client computer 2 is more recent than the metadata 64 stored on the server computer 70. In particular, the request from the productivity application 30 to synchronize the partition 60 may include a timestamp indicating when the metadata 64A was stored in the partition 60 on the client computer 2. The server application 42 may compare the timestamp with a timestamp indicating when the metadata 64 was stored in the partition 60 on the server computer 70. If it is determined that the metadata 64A is more recent than the metadata 64, then the routine 400 continues to operation 415. If it is determined that the metadata 64 is more recent than the metadata 64A, then the routine 400 branches to operation 420. It should be understood by those skilled in the art that, in accordance with another embodiment, the request from the productivity application 30 to synchronize the partition 60 may include a hypertext markup language ("HTTP") entity tag ("ETAG") to identify (and resolve) conflicts.

[0036] At operation 415, the server application 42 automatically synchronizes the partition 60 on the client computer 2 with the server computer 70 to resolve the conflict in favor of the most recent metadata 64A being present on the client computer 2. Thus, the server application 42 may update the metadata 64 in the partition 60 on the server computer 70 with the metadata 64A. From operation 415, the routine 400 returns to operation 320 of FIGURE 3.

[0037] At operation 420, the server application 42 automatically causes the synchronization of the partition 60 on the client computer 2 with the server computer 70 to fail in order to resolve the conflict in favor of the metadata 64 being more recent than the metadata 64A. Thus, the server application 42 may maintain the metadata 64 in the partition 60 on the server computer 70. From operation 420, the routine 400 returns to operation 320 of FIGURE 3.

[0038] FIGURE 5 is a flow diagram illustrating a routine 500 for synchronizing another file partition utilizing a server storage model, in accordance with the embodiment illustrated in FIGURE 3. The routine 500 begins from operation 320 of FIGURE 3 and continues to operation 505, where the server application 42 determines whether a conflict exists with synchronizing the partition 50 on the client computer 2 with the server computer 70. In particular and in accordance with an embodiment, the server application 42 may compare the content 54A stored in the partition 50 on the client computer 2 with the content 54 stored in the partition 50 on the server computer 70. If the content 54

contains edits which are not present in the edited content 54A in the partition 50 on the client computer 2, then the server computer 70 determines there is a conflict and the routine 500 continues to operation 510. If the content 54 does not contain any edits at all, then the routine 500 continues to operation 515.

[0039] At operation 510, the server application 42 generates a conflict error message for the user of the client computer 2. For example, the conflict error message generated by the server application 42 on the client computer 2 may include unseen edits to the content 54 (made, for example by a user of the client computer 6) which have been previously uploaded to the server 70. In response to the conflict error message, a user of the client computer 2 may manually resolve the conflict by merging the edited content 54A with the edits to the content 54, or alternatively, letting the previous edits to the content 54 remain unaltered on the server 70. From operation 510, the routine 500 returns to operation 325 of FIGURE 3.

[0040] At operation 515, the server application 42 synchronizes the edited content 54A in the partition 50 with the server 70 to update the content 54 on the server 70 with the edited content 54A. From operation 515, the routine 500 returns to operation 325 of FIGURE 3.

[0041] FIGURE 6 is a flow diagram illustrating a routine 600 for synchronizing file partitions utilizing a server storage model, in accordance with another embodiment. The routine 600 begins at operation 605, where the productivity application 30 executing on the client computer 2 receives edits to electronic document content (contained in the stream 52) in the partition 50 of the electronic document file 40. In accordance with an embodiment, the electronic document file 40 may be downloaded from the server computer 70 including all of the partitions and streams contained therein. The partition 50 may then be opened in order to edit the content 54 contained therein, thereby creating the content 54A.

[0042] From operation 605, the routine 600 continues to operation 610 where the productivity application 30 generates the metadata 64A in the partition 60 of the electronic document file 40 to identify the editor of the content 54A in the partition 50. In particular, the productivity application 30 may generate metadata to add the name of the editor of the content 54A to the author list 67 in the partition 60 of the electronic document file 40 (thereby generating the author list 67A).

[0043] From operation 610, the routine 600 continues to operation 615 where the productivity application 30 sends a request to the server application 42 to synchronize the partition 60 with the server computer 70 in order to save the changes made to the metadata

64 (i.e., the metadata 64A containing the author list 67A) to the server computer 70. It should be understood that the partition 60 may be synchronized independently from the partition 50 such that only the changed metadata 64A is saved to the server computer 70.

[0044] From operation 615, the routine 600 continues to operation 620 where the productivity application 30 executing on the client computer 6 receives the file partitions 50 and 60 (including the streams 42, 56, 62, and 68 contained therein) from the server computer 70 by downloading the electronic file 40. It should be understood that the client computer 6, in receiving the file partition 60, also receives the metadata 64A identifying the editor of the content 54A from the client computer 2, as a result of the synchronization of the partition 60 on the client computer 2 with the server computer 70 at operation 615. Thus, a user of the client computer 6 is able to determine that a user of the client computer 2 has also downloaded the content 54 from the server computer 70 for editing.

[0045] From operation 620, the routine 600 continues to operation 625 where the productivity application 30 executing on the client computer 6 receives edits to electronic document content (contained in the stream 52) in the partition 50 of the electronic document file 40. The partition 50 may then be opened in order to edit the content 54 contained therein, thereby creating the content 54B. It should be understood that the edited content 54B does not include the edits made on the client computer 2 (i.e., the edited content 54A).

[0046] From operation 625, the routine 600 continues to operation 630 where the productivity application 30 executing on the client computer 6 sends a request to the server application 42 to synchronize the partition 60 with the server computer 70 in order to save the changes made to the metadata 64 (i.e., the metadata 64B containing the author list 67B) to the server computer 70. It should be understood that the partition 60 may be synchronized independently from the partition 50 such that only the changed metadata 64B is saved to the server computer 70.

[0047] From operation 630, the routine 600 continues to operation 635 where the client computer 2 receives the metadata 64B identifying the editor of the content 54B from the client computer 6, as a result of the synchronization of the partition 60 with the server computer 70 at operation 630. In particular, the productivity application 30 executing on the client computer 2 may download the partition 60 from the server computer 70 which has been previously updated with the metadata 64B from the client computer 6. Thus, a user of the client computer 2 is able to determine that a user of the client computer 6 has also downloaded the content 54 from the server computer 70 for editing.

[0048] From operation 635, the routine 600 continues to operation 640 where the productivity application 30 executing on the client computer 2 sends a request to the server application 42 to synchronize the partition 50 with the server computer 70 in order to save the edited content 54A to the server computer 70. It should be understood that the partition 50 may be synchronized independently from the partition 60 such that only the edited content 54A is saved to the server computer 70. It should further be understood that the order of synchronization of the partitions 60 and 50 may be reversed such that the partition 50 is synchronized independently with the server computer 70 prior to the partition 60.

[0049] From operation 640, the routine 600 continues to operation 645 where the productivity application 30 executing on the client computer 6 sends a request to the server application 42 to synchronize the partition 50 with the server computer 70 in order to save the edited content 54B to the server computer 70. It should be understood that the partition 50 may be synchronized independently from the partition 60 such that only the edited content 54B is saved to the server computer 70. It should further be understood that the order of synchronization of the partitions 60 and 50 may be reversed such that the partition 50 is synchronized independently with the server computer 70 prior to the partition 60. An illustrative routine for synchronizing the partition 50 with the server computer 70 will be described in greater detail below with respect to FIGURE 7. From operation 645, the routine 600 continues to operation 600 where it then ends.

[0050] FIGURE 7 is a flow diagram illustrating a routine 700 for synchronizing a file partition on a second client computer utilizing a server storage model, in accordance with the embodiment illustrated in FIGURE 6. The routine 700 begins from operation 645 of FIGURE 6 and continues to operation 605, where the productivity application 30 executing on the client computer 6 checks with the server computer 70 for any updates made to the partition 50. In particular, the productivity application 30 may send a query to the server application 42 as to whether the content 54 in the partition 60 on the server computer 70 has been updated by the client computer 2. As discussed above at operation 620 of FIGURE 6, a user of the client computer 6 may be aware that the content 54 in the partition 50 of the server computer 70 is being edited by another user at the client computer 2 upon receiving the file partitions 50 and 60 in the electronic file 40 downloaded from the server computer 70.

[0051] From operation 705, the routine 700 continues to operation 710 where the client computer 6 receives a confirmation from the server application 42 that the partition 50 on

the server computer 70 has been updated (i.e., by the client computer 2) and thus the edited content 54A from the client computer 2 is available to be merged with the edited content 54B in the partition 50 of the client computer 6. From operation 710, the routine 700 continues to operation 715 where the client computer 6 receives the edited content 54A from the server computer 70.

[0052] From operation 715, the routine 700 continues to operation 720 where the productivity application 30 executing on the client computer 6 requests the server application 42 on the server computer 70 to update the partition 50 with the edited content 54A (from the client computer 2) and the edited content 54B (from the client computer 6). From operation 720, the routine 700 returns to operation 650 of FIGURE 6.

[0053] Although the invention has been described in connection with various illustrative embodiments, those of ordinary skill in the art will understand that many modifications can be made thereto within the scope of the claims that follow. Accordingly, it is not intended that the scope of the invention in any way be limited by the above description, but instead be determined entirely by reference to the claims that follow.

CLAIMS

What is claimed is:

1. A method of synchronizing file partitions utilizing a server storage model in a client-server computer network, comprising:

receiving, at a client computer, edits to content of an electronic document, the content comprising a first stream of a first partition in a file, the file comprising the first partition and a second partition, wherein the first partition and the second partition each include at least one stream;

generating, in the second partition on the client computer, metadata associated with the edits to the electronic document content in the first partition;

synchronizing the second partition on the client computer with a server computer to store the metadata in the second partition of the file on the server computer, wherein the step of synchronizing the second partition comprises:

comparing the metadata generated in the second partition on the client computer with metadata stored in the second partition of the file stored on the server computer,

determining a conflict when the metadata generated in the second partition on the client computer is different than the metadata stored in the second partition of the file on the server computer,

automatically synchronizing the second partition on the client computer with the server computer when the metadata in the second partition on the client computer is more recent than the metadata stored in the second partition of the file stored on the server computer, to resolve the conflict, and

automatically causing the synchronization of the second partition on the client computer with the server computer to fail when the metadata stored in the second partition of the file stored on the server computer is more recent than the metadata stored in the second partition on the client computer, to resolve the conflict; and

synchronizing the first partition on the client computer with the server computer to store the edits made to the electronic document content in a first partition of the file on the server computer.

2. The method of claim 1, wherein synchronizing the first partition on the client computer with the server computer to store the edits made to the electronic document content in a first partition of the file on the server computer comprises:

comparing the edited electronic document content in the first partition on the client computer with electronic document content stored in the first partition of the file stored on the server computer; and

determining a conflict when the electronic document content stored in the first partition of the file stored on the server computer contains edits which are not present in the edited electronic document content in the first partition on the client computer.

3. The method of claim 2 further comprising:

causing the synchronization of the first partition between the client computer and the server computer to fail; and

generating, on the client computer, a conflict error message.

4. The method of claim 1, wherein generating, in the second partition on the client computer, metadata associated with the edits to the electronic document content in the first partition comprises generating, in a stream of the second partition, metadata comprising a list of document authors for the electronic document content.

5. A computer-readable storage medium comprising computer executable instructions which, when executed on a computer, will cause the computer to perform a method of synchronizing file partitions utilizing a server storage model, comprising:

receiving, at a client computer, edits to content of an electronic document, the content comprising a first stream of a first partition in a file, the file comprising at least the first partition and a second partition, each of the at least first and second partitions comprising at least one stream;

generating, in the second partition on the client computer , metadata associated with the edits to the electronic document content in the first partition;

synchronizing the second partition on the client computer with a server computer to store the metadata in a second partition of the file on the server computer, wherein the step of synchronizing the second partition comprises:

comparing the metadata generated in the second partition on the client computer with metadata stored in the second partition of the file stored on the server computer, and

determining a conflict when the metadata generated in the second partition on the client computer is different than the metadata stored in the second partition of the file on the server computer,

automatically synchronizing the second partition on the client computer with the server computer when the metadata in the second partition on the client computer

is more recent than the metadata stored in the second partition of the file stored on the server computer, to resolve the conflict, and

automatically causing the synchronization of the second partition on the client computer with the server computer to fail when the metadata stored in the second partition of the file stored on the server computer is more recent than the metadata stored in the second partition on the client computer, to resolve the conflict; and

synchronizing the first partition on the client computer with the server computer to store the edits made to the electronic document content in a first partition of the file on the server computer (saving the edits made to the electronic document content to the client computer).

6. The computer-readable medium of claim 5, wherein synchronizing the first partition on the client computer with the server computer to store the edits made to the electronic document content in a first partition of the file on the server computer comprises:

comparing the edited electronic document content in the first partition on the client computer with edited electronic document content stored in the first partition of the file stored on the server computer; and

determining a conflict when the electronic document content stored in the first partition of the file stored on the server computer contains edits which are not present in the edited electronic document content in the first partition on the client computer.

7. The computer-readable storage medium of claim 5, wherein generating, in a stream of the second partition on the client computer, metadata associated with the edits to the electronic document content in the first partition comprises generating metadata comprising a list of document authors for the electronic document content.

8. A method of synchronizing file partitions utilizing a server storage model in a client-server computer network, comprising:

receiving, at a first client computer, edits to content of an electronic document, the content comprising a first stream of a first partition in a file, the file comprising at least the first partition and a second partition, wherein the first partition and the second partition each include at least one stream;

generating, in a stream of the second partition on the first client computer, data identifying the editor of the electronic document content in the first partition at the first client computer;

synchronizing the second partition on the first client computer with the server computer to store the data on the server computer;

receiving, at a second client computer, the file stored on the server computer including at least the first partition and the second partition, wherein the second partition comprises the data identifying the editor of the electronic document content at the first client computer and identifying that the electronic document content of the first partition is currently downloaded on the first client computer for editing;

receiving, at the second client computer, edits to the electronic document content in the first partition, wherein the edits received to the electronic document content at the second client computer do not include the edits received to the electronic document content at the first client computer; and

synchronizing the first partition on the first client computer with the server computer to store the edits made to the electronic document content on the server computer.

9. The method of claim 8, wherein generating, in a stream of the second partition on the first client computer, data identifying the editor of the electronic document content in the first partition at the first client computer comprises generating, in a stream of a second partition on the first client computer, metadata identifying the editor of the electronic document content in the first partition at the first client computer.

10. The method of claim 9, wherein synchronizing the second partition on the first client computer with the server computer to store the data on the server computer comprises synchronizing the second partition on the first client computer with the server computer to store the metadata on the server computer.

11. The method of claim 10 further comprising synchronizing the second partition on the second client computer with the server computer to update the metadata identifying the editor of the electronic document content at the first client computer with the identity of the editor of the electronic document content at the second client computer.

12. The method of claim 8 further comprising synchronizing the first partition on the second client computer with the server computer to add, to the first partition on the server computer, the edits made to the electronic document content at the second client computer to the edits made to the electronic document at the first client computer.

13. The method of claim 12, wherein synchronizing the first partition on the second client computer with the server computer to add, to the first partition on the server

computer, the edits made to the electronic document content at the second client computer to the edits made to the electronic document at the first client computer, comprises:

checking the server computer for updates to the first partition; and

receiving a confirmation from the server computer that the first partition has been updated.

14. The method of claim 12, wherein synchronizing the first partition on the second client computer with the server computer to add, to the first partition on the server computer, the edits made to the electronic document content at the second client computer to the edits made to the electronic document at the first client computer, further comprises:

receiving, from the server computer, the edits made to the electronic document content at the first client computer; and

updating the first partition on the server computer with edits made to the electronic document content at the second client computer and the first client computer.