US 20160085920A1

(19) **United States**

(12) **Patent Application Publication**

(10) **Pub. No.: US 2016/0085920 A1**

(43) **Pub. Date: Mar. 24, 2016**

CYRAN

(54) **SYSTEM AND METHOD FOR TRANSFERRING A GROUP OF RELATED FILES AS ONE LOGICAL UNIT**

(71) Applicant: **KONINKLIJKE PHILIPS N.V.**, Eindhoven (NL)

(72) Inventor: **Robert John CYRAN**, Delmont, PA (US)

(21) Appl. No.: **14/890,180**

(22) PCT Filed: **May 28, 2014**

(86) PCT No.: **PCT/IB2014/061768**

§ 371 (c)(1),
(2) Date: **Nov. 10, 2015**

Related U.S. Application Data

(60) Provisional application No. 61/829,306, filed on May 31, 2013.

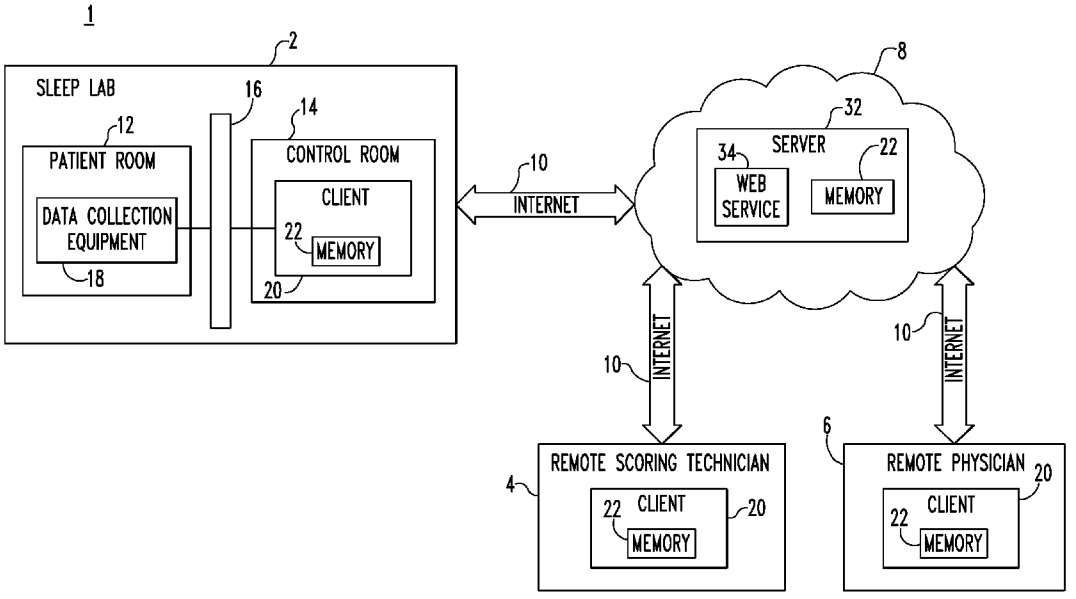
Publication Classification

(51) **Int. Cl.**
G06F 19/00 (2006.01)
H04L 29/08 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 19/322*** (2013.01); ***H04L 67/06*** (2013.01); ***H04L 67/1095*** (2013.01)

(57) **ABSTRACT**

A method of transferring files, the method including receiving a download request for a group of files, determining whether the group of files is locked, upon determining that the group of files is not locked, allowing the download request, creating a transfer object (**40**) corresponding to the group of files and the download request, the transfer object including information on the group of files and the download request, performing download operations on the files in the group of files, and when all files in the group of files have been downloaded, deleting the transfer object.



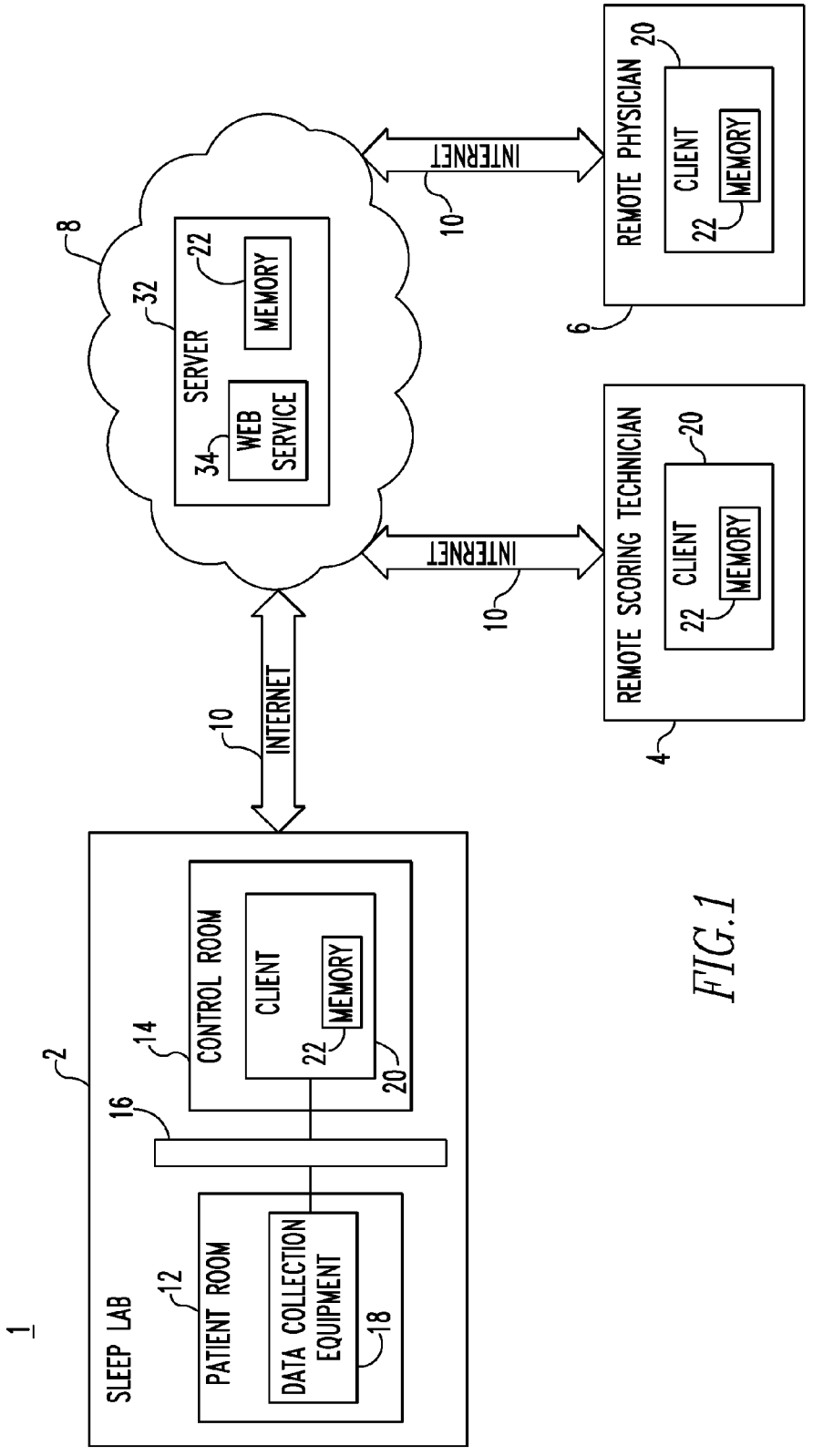

1
SLEEP LAB
2
PATIENT ROOM
12
DATA COLLECTION EQUIPMENT
18
16
CONTROL ROOM
14
CLIENT
MEMORY
22
20
INTERNET
10
8
SERVER
32
MEMORY
22
WEB SERVICE
34
INTERNET
10
REMOTE PHYSICIAN
6
CLIENT
MEMORY
22
20
INTERNET
10
REMOTE SCORING TECHNICIAN
4
CLIENT
MEMORY
22
20

FIG. 1

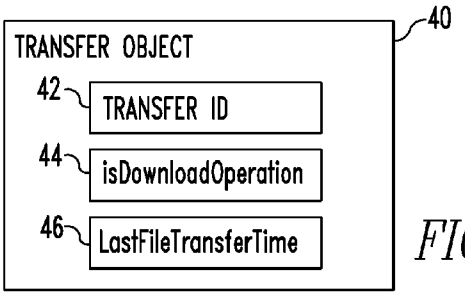

TRANSFER OBJECT
40
42
TRANSFER ID
44
isDownloadOperation
46
LastFileTransferTime

FIG.2

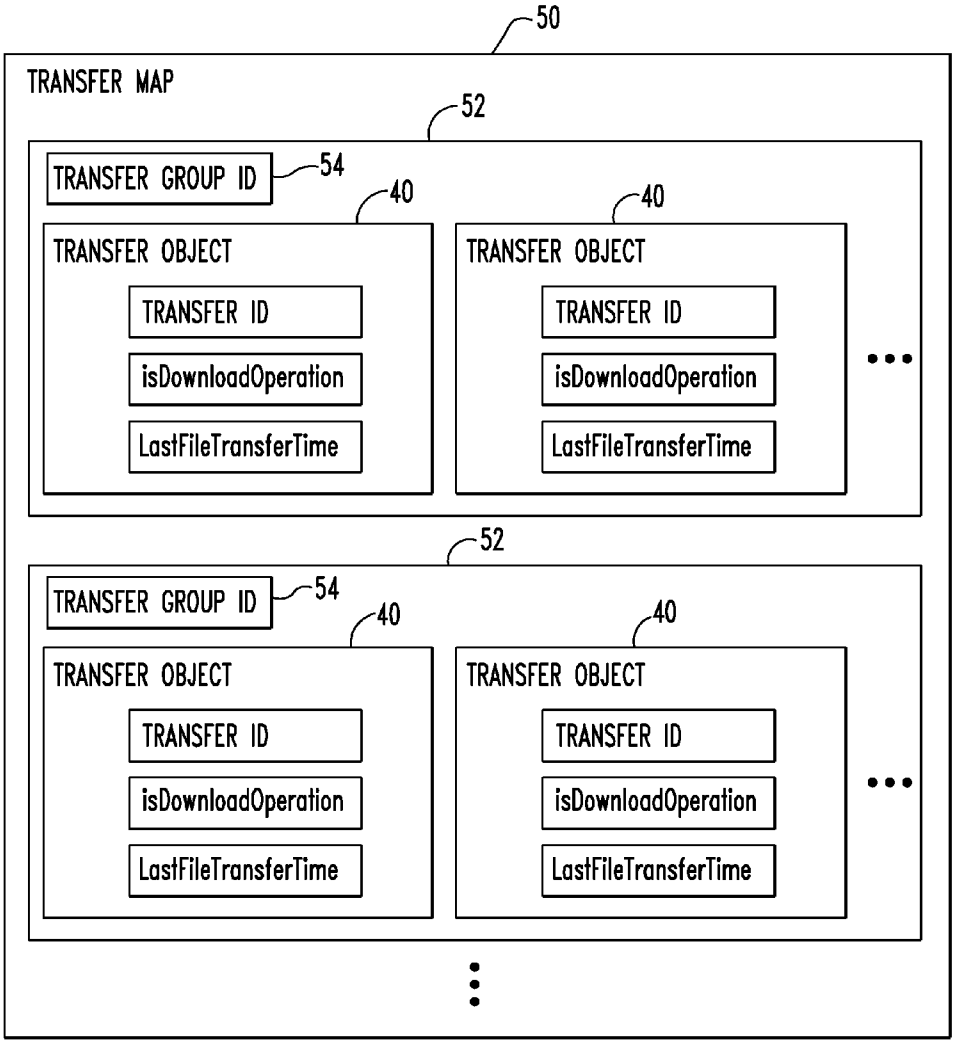

50
TRANSFER MAP
52
TRANSFER GROUP ID
54
40
TRANSFER OBJECT
TRANSFER ID
isDownloadOperation
LastFileTransferTime
40
TRANSFER OBJECT
TRANSFER ID
isDownloadOperation
LastFileTransferTime
52
TRANSFER GROUP ID
54
40
TRANSFER OBJECT
TRANSFER ID
isDownloadOperation
LastFileTransferTime
40
TRANSFER OBJECT
TRANSFER ID
isDownloadOperation
LastFileTransferTime

FIG.3

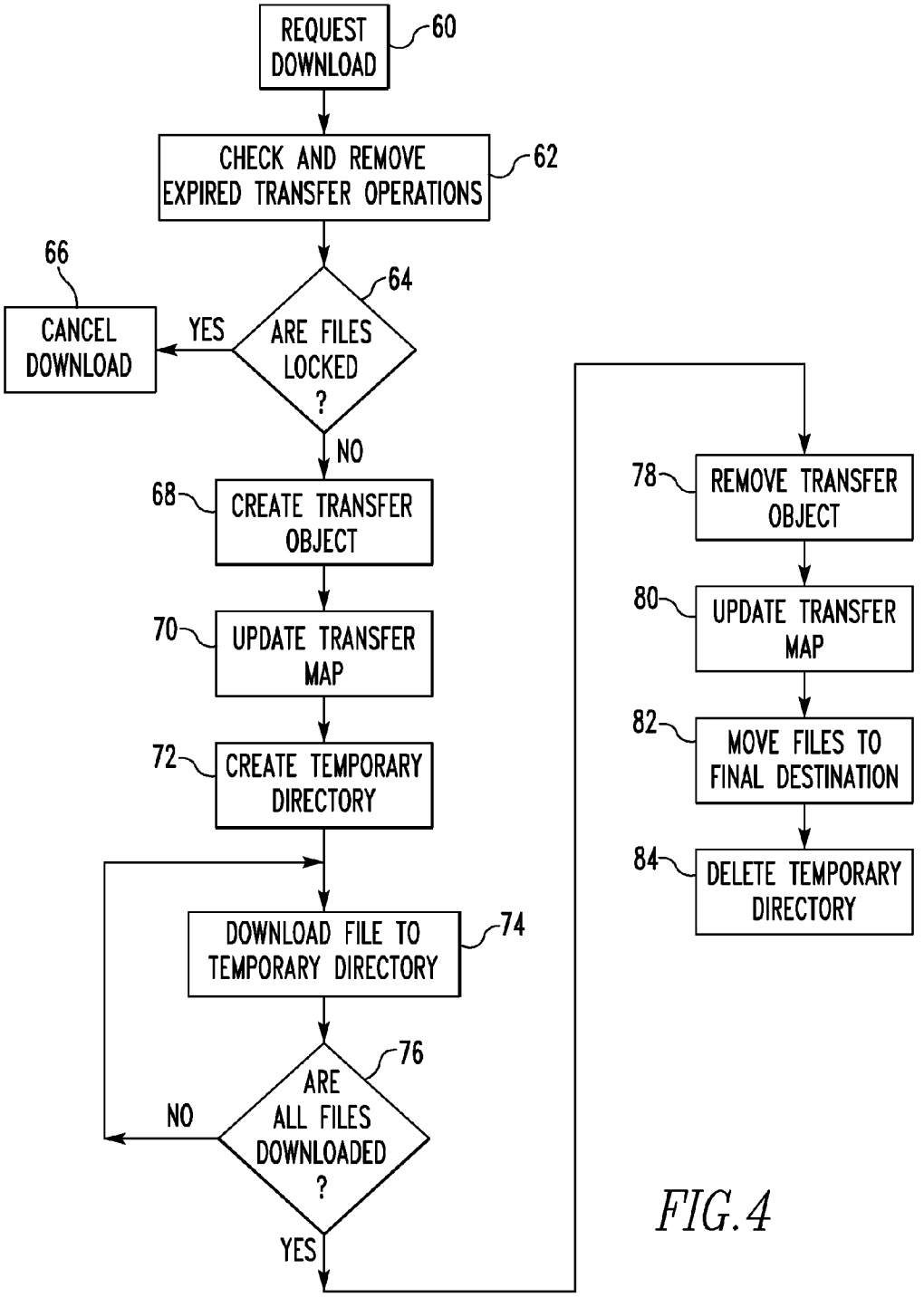

REQUEST DOWNLOAD
60
CHECK AND REMOVE EXPIRED TRANSFER OPERATIONS
62
66
64
CANCEL DOWNLOAD
YES
ARE FILES LOCKED ?
NO
68
CREATE TRANSFER OBJECT
70
UPDATE TRANSFER MAP
72
CREATE TEMPORARY DIRECTORY
DOWNLOAD FILE TO TEMPORARY DIRECTORY
74
76
ARE ALL FILES DOWNLOADED ?
NO
YES
78
REMOVE TRANSFER OBJECT
80
UPDATE TRANSFER MAP
82
MOVE FILES TO FINAL DESTINATION
84
DELETE TEMPORARY DIRECTORY

FIG.4

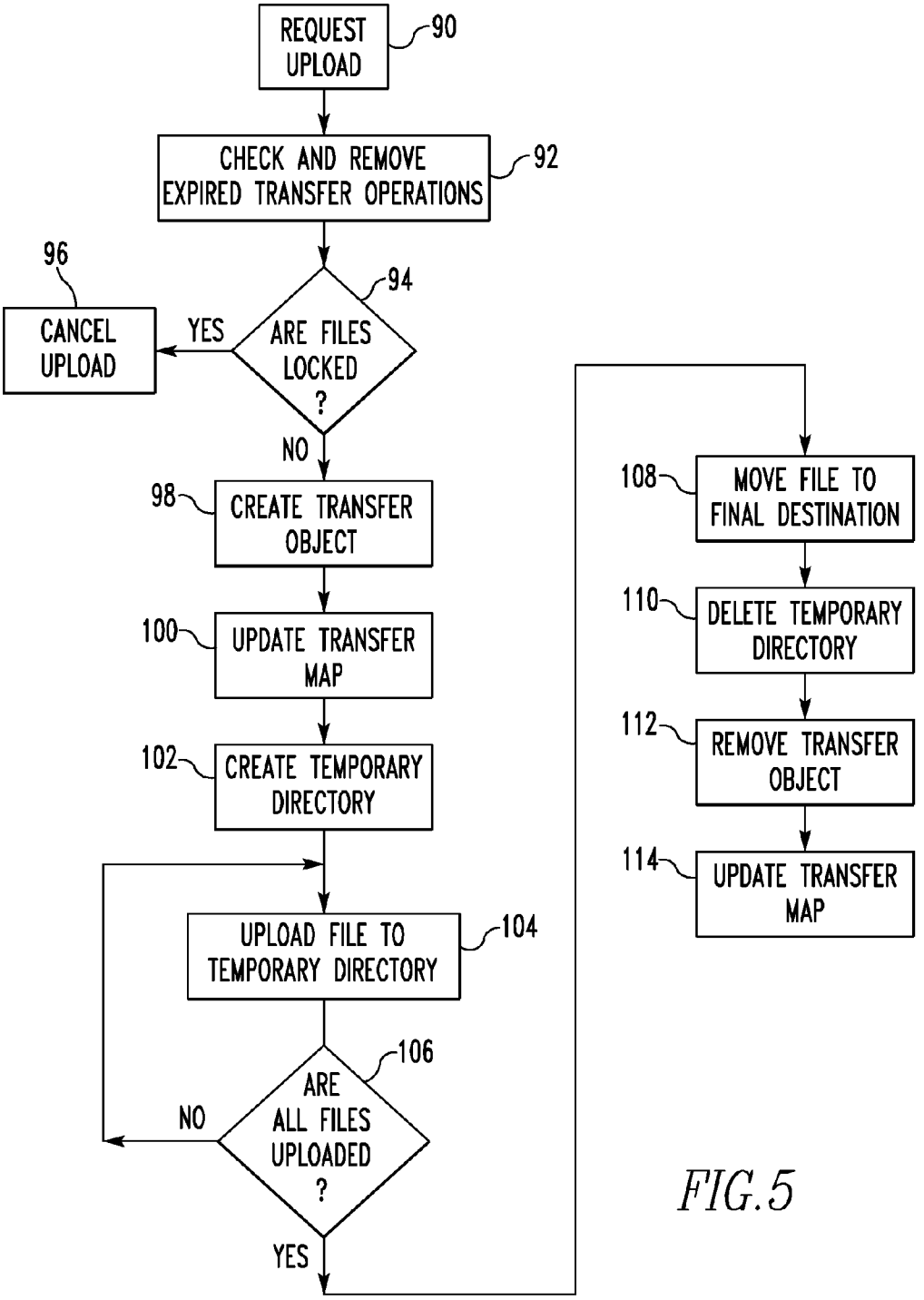

REQUEST UPLOAD
90
CHECK AND REMOVE EXPIRED TRANSFER OPERATIONS
92
ARE FILES LOCKED ?
94
YES
96
CANCEL UPLOAD
NO
98
CREATE TRANSFER OBJECT
100
UPDATE TRANSFER MAP
102
CREATE TEMPORARY DIRECTORY
UPLOAD FILE TO TEMPORARY DIRECTORY
104
ARE ALL FILES UPLOADED ?
106
NO
YES
108
MOVE FILE TO FINAL DESTINATION
110
DELETE TEMPORARY DIRECTORY
112
REMOVE TRANSFER OBJECT
114
UPDATE TRANSFER MAP

FIG.5

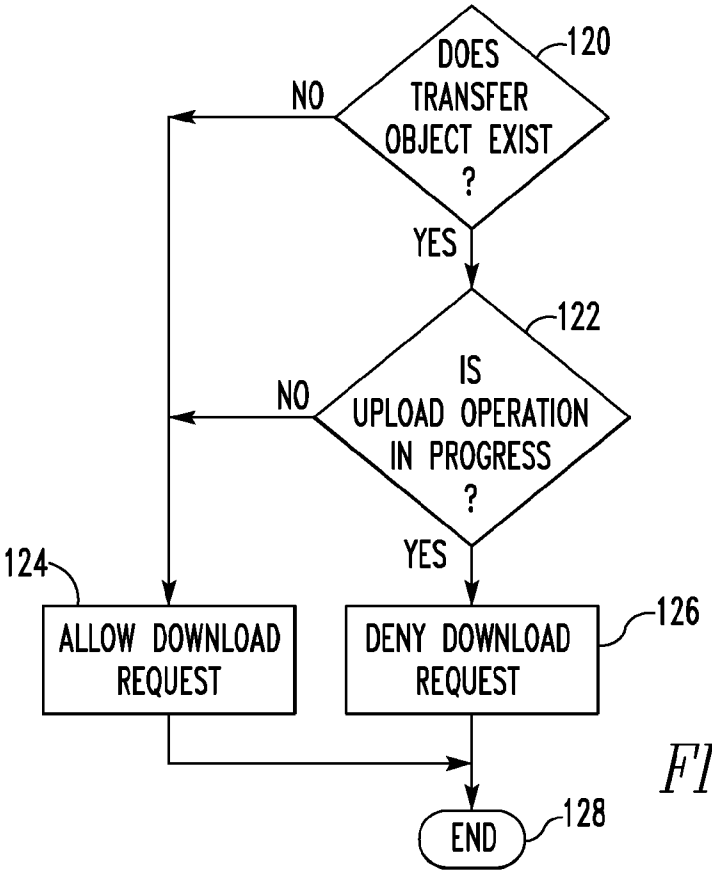

DOES TRANSFER OBJECT EXIST ?
120
NO
YES
IS UPLOAD OPERATION IN PROGRESS ?
122
NO
YES
124
ALLOW DOWNLOAD REQUEST
DENY DOWNLOAD REQUEST
126
END
128

FIG. 6

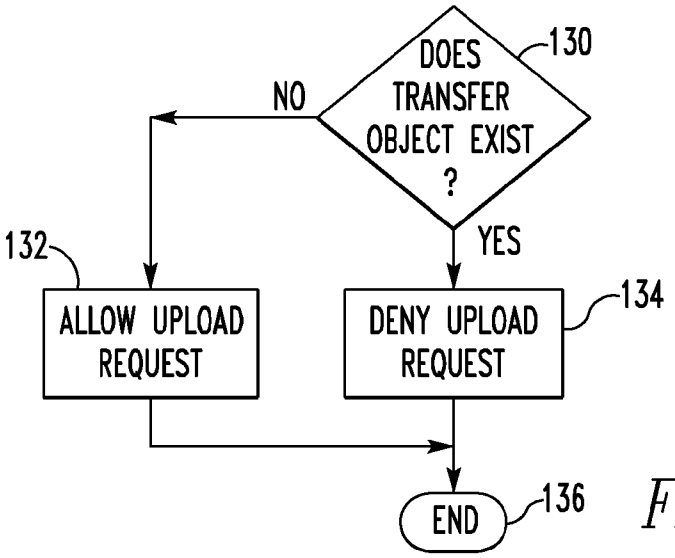

DOES TRANSFER OBJECT EXIST ?
130
NO
YES
132
ALLOW UPLOAD REQUEST
DENY UPLOAD REQUEST
134
END
136

FIG. 7

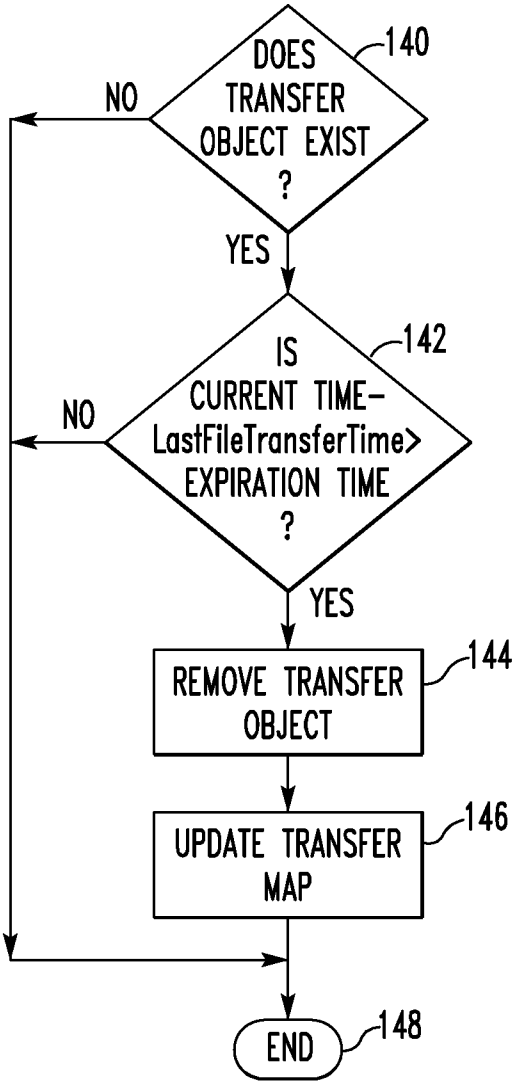

DOES TRANSFER OBJECT EXIST ?
140
NO
YES
IS CURRENT TIME- LastFileTransferTime> EXPIRATION TIME ?
142
NO
YES
REMOVE TRANSFER OBJECT
144
UPDATE TRANSFER MAP
146
END
148

FIG.8

SYSTEM AND METHOD FOR TRANSFERRING A GROUP OF RELATED FILES AS ONE LOGICAL UNIT

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority under 35 U.S.C. §119(e) from U.S. provisional patent application No. 61/829,306, entitled "METHOD AND SYSTEM FOR RELIABLY TRANSFERRING FILES TO AND FROM A REMOTE SERVER" and filed on May 31, 2013, the contents of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

[0002] 1. Field of the Invention

[0003] The present invention pertains to transferring files, and, in particular, to a system and method for transferring sleep study files.

[0004] 2. Description of the Related Art

[0005] Obstructive sleep apnea (OSA) is a condition that affects millions of people from around the world. OSA is characterized by disturbances or cessation in breathing during sleep. OSA episodes result from partial or complete blockage of airflow during sleep that lasts at least 10 seconds and often as long as 1 to 2 minutes. In a given night, people with moderate to severe apnea may experience complete or partial breathing disruptions as high as 200-500 per night. Because their sleep is constantly disrupted, they are deprived of the restorative sleep necessary for efficient functioning of body and mind. This sleep disorder has also been linked with hypertension, depression, stroke, cardiac arrhythmias, myocardial infarction and other cardiovascular disorders. OSA also causes excessive tiredness.

[0006] Various methods have been used to assess whether a patient suffers from OSA. The most comprehensive method is a clinical polysomnogram (PSG), which can diagnose many significant sleep pathologies. A PSG generally involves a sleep study of a patient where audio and other parameters of the patient is recorded while the patient sleeps. A technician then "scores" the recorded data. Scoring the recorded data involves analyzing the data to identify events that occurred during the PSG that may be useful in diagnosing sleep pathologies. Data collected during a PSG is generally stored in several different files such as multiple audio files, other data files, and a scoring file.

[0007] The PSG data (also referred to as sleep study data) is generated or used in several different locations, such as a sleep lab, a scoring technician's office, and a physician's office. It is preferable to store the sleep study data at a central location, such as a server, where it is accessible to multiple locations through a network, such as the internet.

[0008] Current file transfer services upload to a temporary directory on the server, and when the upload is complete, the file is copied to a permanent directory. This protocol prevents an incomplete transfer of a single file, such as one in which connection was lost before the upload completed, from being permanently stored on the server. Some file transfer services forgo the temporary directory and copy files directly to their final destination. However, this process can lead to incomplete files being stored at their final destination in the case of an incomplete transfer.

[0009] When a file is being uploaded, the server will provide synchronization locking which prevents the same file from being uploaded or downloaded by a different user at the same time. However, under current file transfer protocols, uploads are transferred to a permanent directory and synchronization locking is provided on a file-by-file basis. That is, when uploading of one file is complete, the file will be transferred to the permanent directory and the synchronization locking will stop.

[0010] Current file transfer services behave in a similar manner when downloading files from the server to a client computer. That is, they may download a file to a temporary directory to prevent an incomplete file from being downloaded to its final destination directory in the case the connection was lost before the download was complete. Some services may download a file directly to its final destination directory, which can result in an incomplete transfer. Additionally, when a file is being downloaded, the client operating system will provide synchronization locking which prevents other applications on the client from accessing the file while it is being downloaded. However, under current file transfer protocols the synchronization locking is provided on a file-by-file basis.

[0011] In the case of the sleep study data files, file transfers may consist of multiple data files that are related to each other, and as such, must be transferred as one logical unit. In the case of an upload, if the first one of the files is uploaded to the server properly and the connection is lost while uploading the second one of the data files, then the set of sleep study data files stored on the server will be incomplete. Furthermore, if connection is lost while uploading an updated set of sleep study files, the resultant set of sleep study data files on the server will have some data files that are updated and some that are not updated, thus compromising the integrity of the set of sleep study data files stored on the server. Similarly, if a user is uploading the second one of the data files, the synchronization locking for the first data file will have stopped and another user will be able to download the first data file.

[0012] The same issues apply when downloading sleep study files. That is, if the connection to the server is lost when downloading files from the server to the client, then the set of sleep study data files stored on the client will be incomplete. Furthermore, if the connection is lost while downloading an updated set of sleep study files, the resultant set of sleep study data files on the client will have some data files that are updated and some that are not updated, thus compromising the integrity of the set of sleep study data files stored on the client. Similarly, if a user is downloading the first data file, the synchronization locking on the server will not prevent users from updating the other sleep data files on the server while the first is being downloaded. The resultant set of sleep data files downloaded to the client will have some data files that are updated and some that are not updated.

[0013] For the purpose of transferring the sleep study data files to the server, it is preferable to treat the sleep study data files as a single entity to ensure that the complete set of sleep study data files is uploaded to the server before it is transferred to the permanent directory and that the complete set of sleep study data files is synchronization locked when a user is uploading the sleep study data files. Similarly, when downloading sleep study files from the server, it is preferable to treat the sleep study data files as a single entity to ensure that the complete set is downloaded to the client before it is transferred to the permanent directory, and to ensure the complete set of sleep study data files is locked on the server when a user is downloading the study data files. Although in some

cases sleep study data files can be packaged into a single file, such as a ZIP file, and then transferred, the process of packaging and unpacking the sleep study data files can become cumbersome and time consuming.

[0014] Accordingly, a need exists for improvement in transferring sleep study data files to and from a server.

SUMMARY OF THE INVENTION

[0015] In one embodiment, method of transferring files includes receiving a download request for a group of files, determining whether the group of files is locked, upon determining that the group of files is not locked, allowing the download request, creating a transfer object corresponding to the group of files and the download request, the transfer object including information on the group of files and the download request, performing download operations on the files in the group of files, and when all files in the group of files have been downloaded, deleting the transfer object.

[0016] In another embodiment a method of transferring files includes receiving an upload request for a group of files, determining whether the group of files is locked, upon determining that the group of files is not locked, allowing the upload request, creating a transfer object corresponding to the group of files and the upload request, the transfer object including information on the group of files and the upload request, performing upload operations on the files in the group of files, and when all files in the group of files have been uploaded, deleting the transfer object.

[0017] In another embodiment, a non-transitory computer readable medium storing one or more programs, including instructions, which when executed by a computer, causes the computer to perform a method of transferring files, the method including receiving a download request for a group of files, determining whether the group of files is locked, upon determining that the group of files is not locked, allowing the download request, creating a transfer object corresponding to the group of files and the download request, the transfer object including information on the group of files and the download request, performing download operations on the files in the group of files, and when all files in the group of files have been downloaded, deleting the transfer object.

[0018] In another embodiment, a non-transitory computer readable medium storing one or more programs, including instructions, which when executed by a computer, causes the computer to perform a method of transferring files, the method including receiving an upload request for a group of files, determining whether the group of files is locked, upon determining that the group of files is not locked, allowing the upload request, creating a transfer object corresponding to the group of files and the upload request, the transfer object including information on the group of files and the upload request, performing upload operations on the files in the group of files, and when all files in the group of files have been uploaded, deleting the transfer object.

[0019] In another embodiment, a system to transfer files includes a server structured to receive a download request for a group of files, to determine whether the group of files is locked, to allow the download request upon determining that the group of files is not locked, to create a transfer object corresponding to the group of files and the download request, the transfer object including information on the group of files and the download request, to perform download operations on the files in the group of files, and to delete the transfer object when all files in the group of files have been downloaded.

[0020] In another embodiment, a system to transfer files includes a server structured to receive an upload request for a group of files, to determine whether the group of files is locked, to allow the upload request upon determining that the group of files is not locked, to create a transfer object corresponding to the group of files and the upload request, the transfer object including information on the group of files and the upload request, to perform upload operations on the files in the group of files, and to delete the transfer object when all files in the group of files have been uploaded.

[0021] These and other objects, features, and characteristics of the present invention, as well as the methods of operation and functions of the related elements of structure and the combination of parts and economies of manufacture, will become more apparent upon consideration of the following description and the appended claims with reference to the accompanying drawings, all of which form a part of this specification, wherein like reference numerals designate corresponding parts in the various figures. It is to be expressly understood, however, that the drawings are for the purpose of illustration and description only and are not intended as a definition of the limits of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0022] FIG. **1** is a schematic view of a system adapted to collect and store sleep study data in a centralized location according to one exemplary embodiment of the disclosed concept;

[0023] FIG. **2** is a visual representation of a transfer object in accordance with one exemplary embodiment of the disclosed concept;

[0024] FIG. **3** is a visual representation of a transfer map in accordance with one exemplary embodiment of the disclosed concept;

[0025] FIG. **4** is a flowchart of a method of downloading files in accordance with one exemplary embodiment of the disclosed concept;

[0026] FIG. **5** is a flowchart of a method of uploading files in accordance with one exemplary embodiment of the disclosed concept;

[0027] FIGS. **6** and **7** are flowcharts of method of checking if files are locked in accordance with embodiments of the disclosed concept; and

[0028] FIG. **8** is a flowchart of a method of checking for expired transfer operations in accordance with one exemplary embodiment of the disclosed concept.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

[0029] As used herein, the singular form of "a", "an", and "the" include plural references unless the context clearly dictates otherwise. As used herein, the statement that two or more parts or components are "coupled" shall mean that the parts are joined or operate together either directly or indirectly, i.e., through one or more intermediate parts or components, so long as a link occurs. As used herein, "directly coupled" means that two elements are directly in contact with each other. As used herein, "fixedly coupled" or "fixed"

means that two components are coupled so as to move as one while maintaining a constant orientation relative to each other.

[0030] As used herein, the word "unitary" means a component is created as a single piece or unit. That is, a component that includes pieces that are created separately and then coupled together as a unit is not a "unitary" component or body. As employed herein, the statement that two or more parts or components "engage" one another shall mean that the parts exert a force against one another either directly or through one or more intermediate parts or components. As employed herein, the term "number" shall mean one or an integer greater than one (i.e., a plurality).

[0031] Directional phrases used herein, such as, for example and without limitation, top, bottom, left, right, upper, lower, front, back, and derivatives thereof, relate to the orientation of the elements shown in the drawings and are not limiting upon the claims unless expressly recited therein.

[0032] A system **1** adapted to collect and store sleep study data is generally shown in FIG. **1**. System **1** is distributed over several locations including a sleep lab **2**, a remote scoring technician site **4**, a remote physician site **6**, and a central server site **8**. Central server site **8** includes a server **32** and each of sleep lab **2**, remote scoring technician site **4**, and remote physician site **6** include a client **20**. Clients **20** are each communicatively connected to server **32** by suitable network connections such as internet connections **10**.

[0033] Clients **20** are any suitable processing device such as, without limitation, a general-purpose computer, a wireless device, a personal computer, or a mobile phone. Clients **20** and server **32** each include an associated memory **22**. Memory **22** can be any one or more of a variety of types of internal and/or external storage media such as, without limitation, RAM, ROM, EPROM(s), EEPROM(s), FLASH, and the like that provide a storage register, i.e., a machine readable medium, for data storage such as in the fashion of an internal storage area of a computer, and can be volatile memory or nonvolatile memory. Memory **22** may also be located outside of clients **20** or server **32** and communicatively connected with its associated client **20** or server **32**. Memory **22** may also be a removable device that is able to be removed from its associated client **20** or server **32**.

[0034] Sleep lab **2** includes a patient room **12** and a control room **14**. Patient room **12** and control room **14** are communicatively connected by a suitable network connection such as a local area network connection **16**. Patient room **12** includes data collection equipment **18**. Data collection equipment **18** is used to collect raw sleep study data such, without limitation, audio and video data, electroencephalogram (EEG) data, electrocardiogram (ECG) data, electroculogram (EOG) data, electromyogram (EMG) data, measurements of nasal airflow, measurements of blood oxygen levels and/or other physiological parameters. Control room **14** includes client **20** and memory **22**. The raw sleep study data is transferred from data collection equipment **18** to client **20** via local area network connection **16** and is stored in memory **22**. The raw sleep study data is stored in memory **22** as a set of files.

[0035] Remote scoring technician site **4** includes its associated client **20** and memory **22**. A scoring technician at remote scoring technician site **4** "scores" raw sleep study data and stores the scoring data in memory **22** as a file. The scoring data includes data such as, without limitation, sleep staging data, apnea event data, and high heart rate event data.

[0036] Remote physician site **6** includes its associated client **20** and memory **22**. A physician at remote physician site **6** reviews the raw sleep study data and scoring data in order to diagnose a patient.

[0037] Central server site **8** includes server **32** which is communicatively connected with clients **20** in order to send or receive data. Server **32** includes memory **22** which stores the data. Server **32** also includes web service **34**. Web service **34** coordinates communication between server **32** and clients **20** such as responding to requests from clients **20** for data uploads or downloads. Web service **34** also tracks transfer operations between server **32** and clients **20**. In order to track transfer operations between server **32** and clients **20**, web service **34** uses data structures which will be described in more detail hereinafter with respect to FIGS. **2** and **3**. Server **32** is structured to centrally store sleep study data where it can be downloaded or updated by any of clients **20**.

[0038] Although sleep lab **2**, remote scoring technician site **4** and remote physician site **6** are shown in FIG. **1**, it will be appreciated that any suitable remote sites may be adapted to access sleep study data at central server site **8** without departing from the scope of the disclosed concept.

[0039] A visual representation of a transfer object **40** in accordance with an exemplary embodiment of the disclosed concept is generally shown in FIG. **2**. Transfer object **40** is a data structure such as, without limitation, a class that holds information about a transfer operation (e.g., an upload or a download of a group of files) between one client **20** and server **32**. Information held in transfer object **40** includes transfer ID **42**, isDownloadOperation **44**, and lastFileTransferTime **46**. Transfer ID **42** is unique identification information for the transfer operation corresponding to transfer object **40**. IsDownloadOperation **44** is information indicating whether the transfer operation is a download operation (i.e., a transfer from server **32** to client **20**) or an upload operation (i.e., a transfer from client **20** to server **32**). LastFileTransferTime **46** is information indicating the last time a file transfer was started for the transfer operation. LastFileTransferTime **46** is useful for determining whether the transfer operation has expired.

[0040] A visual representation of a transfer map **50** in accordance with an exemplary embodiment of the disclosed concept is generally shown in FIG. **3**. Transfer map **50** is a data structure such as, without limitation, a dictionary or hash table that contains information about all transfer operations that are currently in progress. Transfer map **50** is structured to include file group transfer information **52** corresponding to each sleep study. Each file group transfer information **52** includes a corresponding unique file group ID **54** (e.g., a sleep study ID) that is used to identify the group of files the file group transfer information **52** corresponds to. Each file group transfer information **52** also includes transfer objects **40** for each in progress transfer operation corresponding to the file group ID **54**. The file group transfer information **52** may also include transfer objects for expired transfer operations that have not yet been removed from transfer map **50**.

[0041] Methods of transferring files between clients **20** and server **32** will be described hereinafter with respect to FIGS. **4** and **5**.

[0042] A flowchart of a method of downloading a group of files (e.g., a group of sleep study files) from server **32** to one of clients **20** in accordance with an exemplary embodiment of the disclosed concept is generally shown in FIG. **4**. At **60**, client **20** requests a download from server **32**. Client **20** pro-

vides file group ID **54** to web service **34** in the request for download. Upon receiving the request for download, web service checks for expired transfer operations and removes any expired transfer operations at **62**. The process of checking and removing expired transfer operations will be described in more detail with respect to FIG. **7**.

[0043] After removing expired transfer operations, web service **34** checks if the files requested for download are locked **64**. The process of checking if the requested files are locked will be described in more detail with respect to FIG. **6**. If the files requested for download are locked, web service **34** denies the download request and cancels the download operation at **66**.

[0044] If the requested files are not locked, web service **34** proceeds with the download request and creates a new transfer object **40** at **68**. As previously described, transfer object **40** includes transfer ID **42** which is unique to the transfer operation, isDownloadOperation **44** which holds a value indicating that the current transfer operation is a download, and lastFileTransferTime **46** to hold information on the time the last file was transferred. At **70**, web service **34** updates transfer map **50** to include the recently created transfer object **40**. In the case that transfer map **50** does not include file group transfer information **52** corresponding to the requested group of files, web service **34** creates a new file group transfer information **52** entry including the file group ID **54** corresponding to the requested group of files and includes the recently created transfer object **40** in the new file group transfer information **52**. If file group transfer information **52** corresponding to the requested group of files already exists, web service **34** adds the recently created transfer object **40** to the corresponding file group transfer information **52**. Once transfer map **50** is updated, web service **34** communicates transfer ID **42** back to the client **20** which allows client **20** to begin downloading the requested group of files.

[0045] At **72**, client **20** creates a temporary directory in which to temporarily store the downloaded files. At **74**, client **20** downloads one of the requested files to the temporary directory by communicating file group ID **54**, transfer ID **42** and the name of the requested file to web service **34**. Web service **34** locates the requested file and begins the download of the requested file. Web service **34** also updates the lastFileTransferTime **46** in the transfer object **40** corresponding to the current operation to indicate the time that the download operation started. Upon receiving the requested file, client **20** places the file in the temporary directory. At **76**, client **20** checks if all files in the requested group of files are downloaded. If all the files are not downloaded, client **20** repeats **74** for the next file in the requested group of files. If all the files are downloaded, client **20** informs web service **34** to indicate that the download operation is complete.

[0046] At **78**, web service **34** removes the transfer object **40** corresponding to the recently completed download operation. At **80**, web service **34** updates transfer map **50** by removing transfer object **40** from transfer map **50** and, if there are no transfer objects **40** remaining in the file group transfer information **52** corresponding to the requested group of files, web service **34** also removes the file group transfer information **52** corresponding to the requested group of files from transfer map **50**.

[0047] At **82**, client **20** moves the downloaded files from the temporary directory to their final destination, such as a directory where a user requested the files to be downloaded to. At **84**, client **20** deletes the temporary directory.

[0048] A flowchart of a method of uploading a group of files (e.g., a group of sleep study files) from one of clients **20** to server **32** in accordance with an exemplary embodiment of the disclosed concept is generally shown in FIG. **5**. At **90**, client **20** requests an upload to server **32**. Client **20** provides file group ID **54** to web service **34** in the request for upload. Upon receiving the request for upload, web service **34** checks for expired transfer operations and removes any expired transfer operations at **92**. The process of checking and removing expired transfer operations will be described in more detail with respect to FIG. **7**.

[0049] After removing expired transfer operations, web service **34** checks if the files requested for upload are locked **94**. The process of checking if the requested files are locked will be described in more detail with respect to FIG. **6**. If the files requested for upload are locked, web service **34** denies the upload request and cancels the upload operation at **96**.

[0050] If the requested files are not locked, web service **34** proceeds with the upload request and creates a new transfer object **40** at **98**. As previously described, transfer object **40** includes transfer ID **42** which is unique to the transfer operation, isDownloadOperation **44** which holds a value indicating that the current transfer operation is an upload, and lastFileTransferTime **46** to hold information on the time the last file was transferred. At **100**, web service **34** updates transfer map **50** to include the recently created transfer object **40**. In the case that transfer map **50** does not include file group transfer information **52** corresponding to the requested group of files, web service **34** creates a new file group transfer information **52** entry including the file group ID **54** corresponding to the requested group of files and includes the recently created transfer object **40** in the new file group transfer information **52**. If file group transfer information **52** corresponding to the requested group of files already exists, web service **34** adds the recently created transfer object **40** to the corresponding file group transfer information **52**.

[0051] At **102**, web service **34** creates a temporary directory on server **32** to store uploaded files. Once the temporary directory is created, web service **34** communicates transfer ID **42** back to the client **20** which allows client **20** to begin uploading the requested group of files.

[0052] At **104**, client **20** uploads one of the requested files to the temporary directory on server **32**. When a file is uploaded, client **20** communicates file group ID **54**, transfer ID **42** and the name of the requested file to web service **34**. Web service **34** updates the lastFileTransferTime **46** in the transfer object **40** corresponding to the current operation to indicate the time that the upload operation started. Upon receiving the requested file, web service **34** places the file in the temporary directory on server **32**. At **106**, client **20** checks if all files in the requested group of files are uploaded. If all the files are not uploaded, client **20** repeats **104** for the next file in the requested group of files. If all the files are uploaded, client **20** informs web service **34** to indicate that the download operation is complete.

[0053] At **108**, web service **34** moves the uploaded files from the temporary directory to their final destination. At **110**, web service **34** deletes the temporary directory.

[0054] At **112**, web service **34** removes the transfer object **40** corresponding to the recently completed upload operation. At **114**, web service **34** updates transfer map **50** by removing transfer object **40** from transfer map **50** and, if there are no transfer objects **40** remaining in the file group transfer information **52** corresponding to the requested group of files, web

service **34** also removes the file group transfer information **52** corresponding to the requested group of files from transfer map **50**.

[0055] A flowchart of a method of checking if files are locked from a download operation in accordance with an exemplary embodiment of the disclosed concept is generally shown in FIG. **6**. This method is used by web service **34** to check if a group of files is locked from being downloaded at **64** in FIG. **4**. At **120**, web service **34** checks if a transfer object **40** exists for the requested group of files. Web service **34** checks transfer map **50** to see if there is file group transfer information **52** corresponding to the file group ID **54** for the requested group of files. If the file group transfer information **52** is not present in transfer map **50**, then the web service **34** determines that no transfer operations are in progress for the requested group of files and web service **34** proceeds to allow the download request at **124**.

[0056] If the file group transfer information **52** is present in transfer map **50**, web service **34** checks all transfer objects **40** in key to determine whether any upload operations are in progress for the requested group of files. Web service **34** checks isDownloadOperation **44** in the file transfer objects **40** to determine whether any uploads are in progress for the requested group of files. Multiple simultaneous downloads are permitted for a group of files, so if web service **34** determines that no upload operations are in progress, web service **34** proceeds to allow the download operation at **124**. However, if there is an upload operation in progress, web service **34** proceeds to deny the download request at **126** and informs client **20** that the download request has been denied. At **128**, the method of FIG. **6** ends. After the method of FIG. **6** ends, the method of FIG. **4** continues at **66** if the download request has been denied or at **68** if the download request has been allowed.

[0057] A flowchart of a method of checking if files are locked from an upload operation in accordance with an exemplary embodiment of the disclosed concept is generally shown in FIG. **7**. This method is used by web service **34** to check if a group of files is locked from being uploaded at **94** in FIG. **5**. At **130**, web service **34** checks if a transfer object **40** exists for the requested group of files. Web service **34** checks transfer map **50** to see if there is file group transfer information **52** corresponding to the file group ID **54** for the requested group of files. If the file group transfer information **52** is not present in transfer map **50**, then the web service **34** determines that no transfer operations are in progress for the requested group of files and web service **34** proceeds to allow the upload request at **132**. If the file group transfer information **52** is present in transfer map **50**, web service **34** determines that a download or upload operation is currently in progress and proceeds to deny the upload request at **134**.

[0058] At **136**, the method of FIG. **7** ends. After the method of FIG. **7** ends, the method of FIG. **5** continues at **96** if the upload request has been denied or at **98** if the upload request has been allowed.

[0059] A flowchart of a method of checking and removing expired transfer operations in accordance with an exemplary embodiment of the disclosed concept is generally shown in FIG. **8**. This method is used by web service **34** to check and remove expired transfer operations at **62** in FIGS. **4** and **92** in FIG. **5**. At **140**, web service **34** checks if a transfer object **40** exists for the requested group of files. Web service **34** checks transfer map **50** to see if there is file group transfer information **52** corresponding to the file group ID **54** for the requested group of files. If the file group transfer information **52** is not present in transfer map **50**, then the web service **34** determines that there are no expired transfer operations and proceeds to end the routine at **148**.

[0060] If the file group transfer information **52** is present, web service **34** proceeds to check the transfer objects **40** in the file group transfer information **52** to determine whether any transfer operations are expired at **142**. Expired file transfers are those that have been in progress for a time that exceeds a predetermined expiration time for an individual file, and that won't complete because the transfer has failed for any reason such as, without limitation, a loss of internet connection. Web service **34** checks lastFileTransferTime **46** in the transfer objects **40** in file group transfer information **52** to determine whether the time between when the last file transfer started and the current time exceeds a predetermined expiration time. The predetermined expiration time is a selected maximum amount of time a single file transfer is permitted to take. If the time between when the last file transfer started and the current time does not exceed the predetermined expiration time, web service **34** determines that the transfer operation is still in progress and proceeds to end the routine at **148**. If the time between when the last file transfer started and the current time exceeds the predetermined expiration time, web service **34** determines that the transfer operation meeting this condition has expired and proceeds to remove the corresponding transfer object **40** at **144**. Web service **34** then updates transfer map **50** at **146** to remove transfer object **40** from transfer map **50**. After, the method of FIG. **8** ends at **148**, the method of FIG. **4** continues at **64** or the method of FIG. **5** continues at **94**.

[0061] Setting the predetermined expiration time for an individual file transfer allows web service **34** to clean up transfer objects **40** and temporary directories for transfer operations that have failed for any reason such as, without limitation, a loss of internet connection. Clean up of transfer object **40** prevents files from being locked indefinitely, which in turn will allow other users to perform file transfer operations on the files in the file group. The predetermined expiration time also aids in preventing Denial of Service (DoS) attacks on the server **32** by limiting the amount of time for a single file transfer.

[0062] Although the disclosed concept has been described in the context of transferring sleep study data, it is contemplated that the disclosed concept may also be adapted for use in transferring any type of data between remote locations and a central storage location. It will be appreciated that the disclosed concept is particularly suitable for use in transferring groups of files.

[0063] The disclosed concept can also be embodied as computer readable codes on a tangible, non-transitory computer readable recording medium. The computer readable recording medium is any data storage device that can store data which can be thereafter read by a computer system. Non-limiting examples of the computer readable recording medium include read-only memory (ROM), non-volatile random-access memory (RAM), CD-ROMs, magnetic tapes, floppy disks, disk storage devices, and optical data storage devices.

[0064] In the claims, any reference signs placed between parentheses shall not be construed as limiting the claim. The word "comprising" or "including" does not exclude the presence of elements or steps other than those listed in a claim. In a device claim enumerating several means, several of these means may be embodied by one and the same item of hard-

ware. The word "a" or "an" preceding an element does not exclude the presence of a plurality of such elements. In any device claim enumerating several means, several of these means may be embodied by one and the same item of hardware. The mere fact that certain elements are recited in mutually different dependent claims does not indicate that these elements cannot be used in combination.

[0065] Although the invention has been described in detail for the purpose of illustration based on what is currently considered to be the most practical and preferred embodiments, it is to be understood that such detail is solely for that purpose and that the invention is not limited to the disclosed embodiments, but, on the contrary, is intended to cover modifications and equivalent arrangements that are within the spirit and scope of the appended claims. For example, it is to be understood that the present invention contemplates that, to the extent possible, one or more features of any embodiment can be combined with one or more features of any other embodiment.

1. A method of transferring files, the method comprising:
receiving a download request for a group of files;
determining whether the group of files is locked;
upon determining that the group of files is not locked, allowing the download request;
creating a transfer object corresponding to the group of files and the download request, the transfer object including information on the group of files and the download request;
performing download operations on the files in the group of files; and
when all files in the group of files have been downloaded, deleting the transfer object,
wherein determining whether the group of files is not locked comprises:
prior to creating the transfer object, determining whether other transfer objects corresponding to the group of files exists;
upon determining that other transfer objects corresponding to the group of files exists, determining for each other transfer object whether the other transfer object corresponds to an upload operation; and
determining that the group of files is not locked and allowing the download request if other transfer objects corresponding to the group of files do not exist or none of the other transfer objects correspond to an upload operation.

2. The method of claim **1**, further comprising:
checking for expired transfer operations; and
removing expired transfer operations.

3. The method of claim **2**, wherein checking for expired transfer operations comprises:
prior to creating the transfer object, determining whether other transfer objects corresponding to the group of files exist;
upon determining that other transfer objects corresponding to the group of files exist, determining for each other transfer object whether a current time minus a last transfer time information included in the other transfer object is greater than a predetermined expiration time; and
in response to determining that the current time minus the last transfer time information included in the other transfer object is greater than a predetermined expiration time, deleting the other transfer object.

4. (canceled)

5. The method of claim **1**, further comprising:
upon creating the transfer object, adding the transfer object to a transfer map, the transfer map including transfer objects corresponding to the group of files and other groups of files; and
when all files in the group of files have been downloaded, removing the transfer object from the transfer map.

6. The method of claim **1**, further comprising:
creating a temporary directory;
downloading files from the group of files to the temporary directory; and
when all files in the group of files have been downloaded, moving the group of files from the temporary directory to a final directory and deleting the temporary directory.

7-14. (canceled)

15. A non-transitory computer readable medium storing one or more programs, including instructions, which when executed by a computer, causes the computer to perform a method of transferring files, the method comprising:
receiving a download request for a group of files;
determining whether the group of files is locked;
upon determining that the group of files is not locked, allowing the download request;
creating a transfer object corresponding to the group of files and the download request, the transfer object including information on the group of files and the download request;
performing download operations on the files in the group of files; and
when all files in the group of files have been downloaded, deleting the transfer object
wherein determining whether the group of files is unlocked comprises:
prior to creating the transfer object, determining whether other transfer objects corresponding to the group of files exists;
upon determining that other transfer objects corresponding to the group of files exists, determining for each other transfer object whether the other transfer object corresponds to an upload operation; and
determining that the group of files is not locked and allowing the download request if other transfer objects corresponding to the group of files do not exist or none of the other transfer objects correspond to an upload operation.

16. The non-transitory computer readable medium of claim **15**, wherein the method further comprises:
checking for expired transfer operations; and
removing expired transfer operations.

17. The non-transitory computer readable medium of claim **16**, wherein checking for expired transfer operations comprises:
prior to creating the transfer object, determining whether other transfer objects corresponding to the group of files exist;
upon determining that other transfer objects corresponding to the group of files exist, determining for each other transfer object whether a current time minus a last transfer time information included in the other transfer object is greater than a predetermined expiration time; and
in response to determining that the current time minus the last transfer time information included in the other transfer object is greater than a predetermined expiration time, deleting the other transfer object.

18. (canceled)

19. The non-transitory computer readable medium of claim **15**, wherein the method further comprises:

upon creating the transfer object, adding the transfer object to a transfer map, the transfer map including transfer objects corresponding to the group of files and other groups of files; and

when all files in the group of files have been downloaded, removing the transfer object from the transfer map.

20. The non-transitory computer readable medium of claim **15**, wherein the method further comprises:

creating a temporary directory;

downloading files from the group of files to the temporary directory; and

when all files in the group of files have been downloaded, moving the group of files from the temporary directory to a final directory and deleting the temporary directory.

21-28. (canceled)

29. A system to transfer files, the system comprising:

a server structured to receive a download request for a group of files, to determine whether the group of files is locked, to allow the download request upon determining that the group of files is not locked, to create a transfer object corresponding to the group of files and the download request, the transfer object including information on the group of files and the download request, to perform download operations on the files in the group of files, and to delete the transfer object when all files in the group of files have been downloaded,

wherein the server is structured to determine whether other transfer objects corresponding to the group of files exists prior to creating the transfer object, wherein, upon determining that other transfer objects corresponding to the group of files exists, the server is structured to determine for each other transfer object whether the other transfer object corresponds to an upload operation, and wherein the server is structured to determine that the group of files is not locked and to allow the download request if other transfer objects corresponding to the group of files do not exist or none of the other transfer objects correspond to an upload operation.

30. The system of claim **29**, wherein the server is structured to check for expired transfer operations and to remove expired transfer operations.

31. The system of claim **30**, wherein the server is structured to determine whether other transfer objects corresponding to the group of files exists prior to creating the transfer object, wherein, upon determining that other transfer objects corresponding to the group of files exist, the server is structured to determine for each other transfer object whether a current time minus a last transfer time information included in the other transfer object is greater than a predetermined expiration time, wherein the server is structured to delete the other transfer object in response to determining that the current time minus the last transfer time information included in the other transfer object is greater than a predetermined expiration time.

32. (canceled)

33. The system of claim **29**, wherein the server is structured to add the transfer object to a transfer map (**50**) upon creating the transfer object, wherein the transfer map includes transfer objects corresponding to the group of files and other groups of files, and wherein the server is structured to remove the transfer object from the transfer map when all files in the group of files have been downloaded.

34. The system of claim **29**, further comprising:

a client device (**20**) structured to create a temporary directory, to download files from the group of files from the server to the temporary directory, and to move the group of files from the temporary directory to a final directory and to delete the temporary directory when all files in the group of files have been downloaded.

35-42. (canceled)

* * * * *