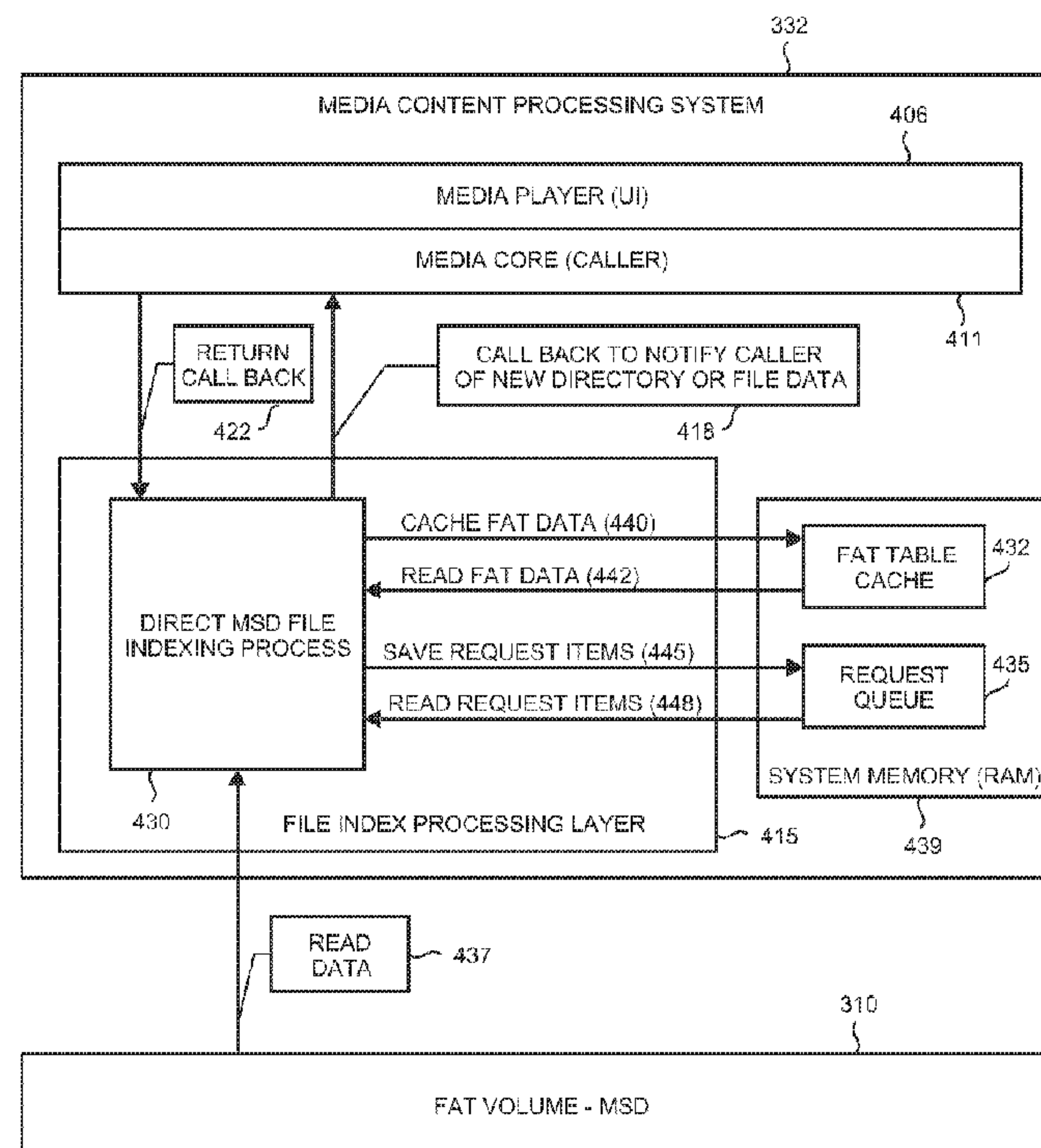




(86) Date de dépôt PCT/PCT Filing Date: 2008/08/20
(87) Date publication PCT/PCT Publication Date: 2009/03/05
(45) Date de délivrance/Issue Date: 2015/11/03
(85) Entrée phase nationale/National Entry: 2010/01/11
(86) N° demande PCT/PCT Application No.: US 2008/073766
(87) N° publication PCT/PCT Publication No.: 2009/029460
(30) Priorités/Priorities: 2007/08/24 (US60/966,032);
2008/01/23 (US12/018,207)

(51) Cl.Int./Int.Cl. *G11B 20/10* (2006.01),
G06F 12/00 (2006.01)
(72) Inventeurs/Inventors:
WHITEHORN, JASON, US;
HENDRIXSON, CORY, US;
LEE, YEN-TSANG, US
(73) Propriétaire/Owner:
MICROSOFT TECHNOLOGY LICENSING, LLC, US
(74) Agent: SMART & BIGGAR

(54) Titre : INDEXATION DE FICHIERS D'UN DISPOSITIF DE MEMOIRE DE MASSE A ACCES SELECTIF
(54) Title: DIRECT MASS STORAGE DEVICE FILE INDEXING



(57) **Abrégé/Abstract:**

An arrangement for enumerating data, such as media content including music, that is stored on external hard drive-based mass storage devices is provided by a media content processing system that implements a direct mass storage device file indexing process. This file indexing process is configured for finding all files and directories on the mass storage device, and reading through those parts of the files which contain metadata (such as album name, artist name, genre, track title, track number etc.) about the file. Use of the media content processing system reduces file enumeration time by minimizing the amount of physical movement of the read/write head in the hard disk drive that is used by the mass storage device. This motion minimization is accomplished by reading the clusters of directory and file data in a sequential manner on the hard disk, rather than randomly performing such read operations.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
5 March 2009 (05.03.2009)

PCT

(10) International Publication Number
WO 2009/029460 A3

(51) International Patent Classification:

G11B 20/10 (2006.01) **G06F 12/00** (2006.01)

(21) International Application Number:

PCT/US2008/073766

(22) International Filing Date: 20 August 2008 (20.08.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/966,032 24 August 2007 (24.08.2007) US
12/018,207 23 January 2008 (23.01.2008) US(71) Applicant (for all designated States except US): **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).(72) Inventors: **WHITEHORN, Jason**; c/o Microsoft Corporation International Patents, One Microsoft Way, Redmond,Washington 98052-6399 (US). **HENDRIXSON, Cory**; c/o Microsoft Corporation International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **LEE, Yen-tsang**; c/o Microsoft Corporation International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

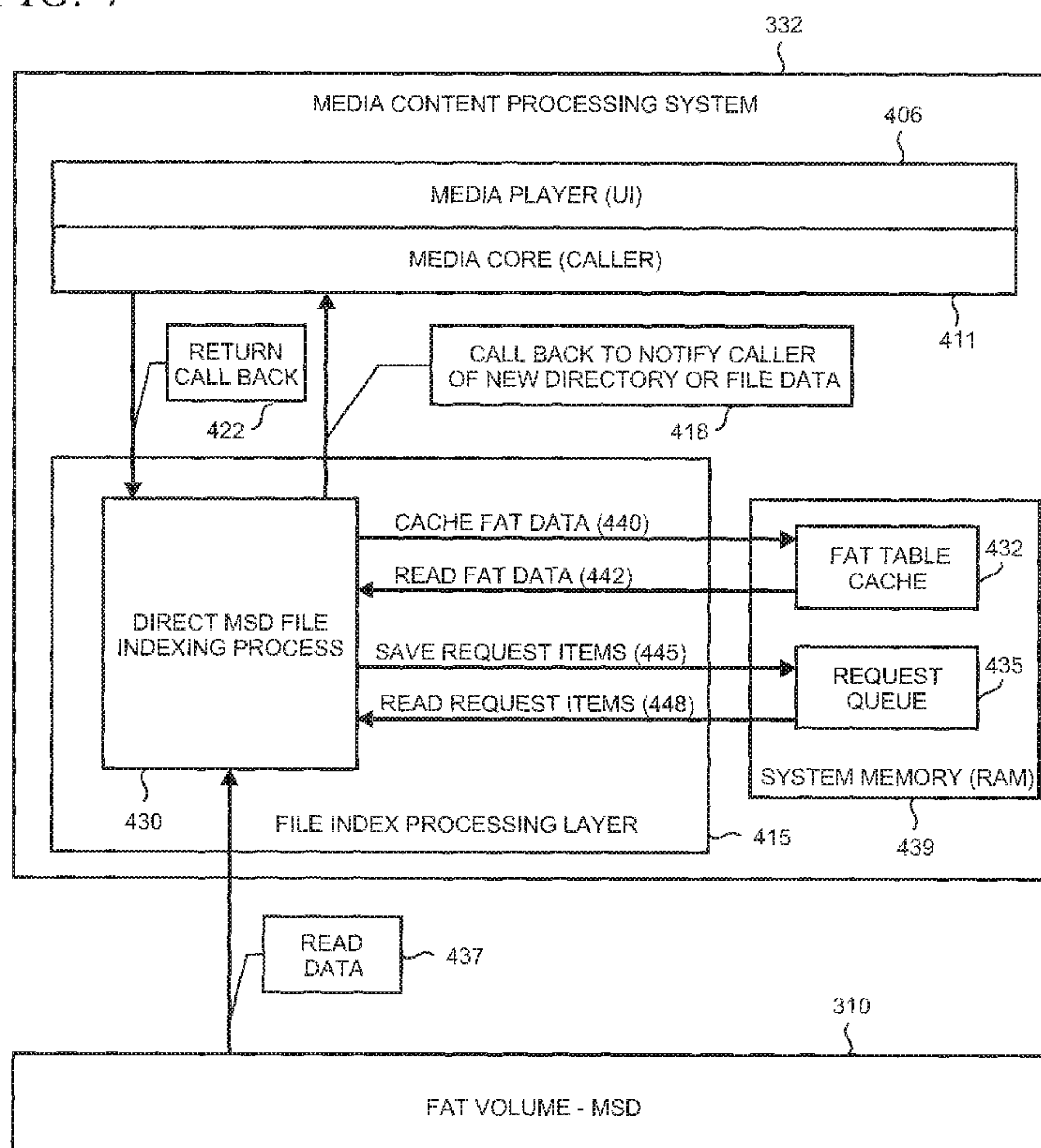
(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,

[Continued on next page]

(54) Title: DIRECT MASS STORAGE DEVICE FILE INDEXING

FIG. 4



(57) Abstract: An arrangement for enumerating data, such as media content including music, that is stored on external hard drive-based mass storage devices is provided by a media content processing system that implements a direct mass storage device file indexing process. This file indexing process is configured for finding all files and directories on the mass storage device, and reading through those parts of the files which contain metadata (such as album name, artist name, genre, track title, track number etc.) about the file. Use of the media content processing system reduces file enumeration time by minimizing the amount of physical movement of the read/write head in the hard disk drive that is used by the mass storage device. This motion minimization is accomplished by reading the clusters of directory and file data in a sequential manner on the hard disk, rather than randomly performing such read operations.

WO 2009/029460 A3



ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

(88) Date of publication of the international search report:
30 April 2009

DIRECT MASS STORAGE DEVICE FILE INDEXING

BACKGROUND

[0001] User demand for mass storage capacity continues to grow, especially for storing large audio, video, image, and multimedia files. This capacity demand has affected the design and development of hard disks and removable media such as CDs (compact discs) and DVDs (digital versatile discs). Storage technologies are further evolving to meet user demands for increasingly greater capacity and more flexible capabilities. Examples of such technologies include compact and portable mass storage devices. Mass storage devices are a class of devices used for storing data in a volume which can be shared with other devices and resources using a data transfer protocol running, for example, on a high speed external bus such as Universal Serial Bus (“USB”) or IEEE-1394 (Institute of Electrical and Electronics Engineers).

[0002] While some mass storage devices use solid state memory as a storage medium, larger capacity portable mass storage devices typically use a small-sized hard disk drive that may often be powered through the USB or IEEE-1394 data cable itself rather than use a separate power cord. These disk-based mass storage devices can thus enable plug-and-play convenience for users with a compact form factor while providing very large amounts of storage for multimedia including, for example, pictures and music libraries.

[0003] Mass storage devices typically store data in the form of files which are organized using a file system. The FAT (file allocation table) file system is one commonly used file system for disk-based mass storage devices. The FAT file system has its origins in the late 1970s and early 1980s and was the file system supported by the Microsoft MS-DOS operating system. It was originally developed as a simple file system suitable for floppy disk drives less than 500K (kilobytes) in size. Over time it has been enhanced to support larger and larger media. Currently, there are three FAT file system types: FAT12, FAT16, and FAT32. The basic difference in these FAT sub types, and the reason for the names, is the size, in bits, of the entries in the actual FAT structure on the disk. There are 12 bits in a FAT12 FAT entry, 16 bits in a FAT16 FAT entry, and 32 bits in a FAT32 FAT entry.

[0004] The FAT file system is characterized by the file allocation table (the “FAT”), which is really a table that resides in a reserved portion of volume. To protect the volume, two copies of the FAT are kept in case one becomes damaged. The FAT tables and the root directory are also stored in a fixed location so that the system’s boot files can be correctly located.

[0005] While the FAT file system performs well in many applications, it has some inherent limitations. In particular, there is no organization to the FAT directory structure, and files and directories are written to the first open location on a disk. As a result, the clusters used for the files and directories can be randomly distributed on the disk in locations that are not logically close to one another. Accessing the data to enumerate a file index for the volume’s contents can be undesirably time consuming because the hard disk drive read/write head must constantly move back and forth, to and from the different tracks on the disk, as it reads the relevant clusters.

[0006] This Background is provided to introduce a brief context for the Summary and Detailed Description that follow. This Background is not intended to be an aid in determining the scope of the claimed subject matter nor be viewed as limiting the claimed subject matter to implementations that solve any or all of the disadvantages or problems presented above.

SUMMARY

[0007] An arrangement for enumerating data, such as media content including music, that is stored on external hard disk drive-based mass storage devices is provided by a media content processing system that implements a direct mass storage device file indexing process. This file indexing process is configured for finding all files and directories on the mass storage device, and reading through those parts of the files which contain metadata (such as album name, artist name, genre, track title, track number, etc.) about the file.

[0008] Use of the media content processing system reduces file enumeration time by minimizing the amount of physical movement of the read/write head in the mass storage device’s hard disk drive as it reads data from the disk. This motion minimization is accomplished by reading the clusters of directory and file data in a sequential manner from the hard disk, rather than by randomly performing such read operations. The media content processing system keeps track of the location of clusters it must process in a work list (i.e., a request queue). Items in the request queue are processed by selecting the next closest cluster to the current physical location of the hard drive read/write head. If additional

51331-868

clusters are required to process an item, those clusters are added to the request queue and processed later, for example, in a subsequent iteration of the direct mass storage indexing process.

[0010] 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 to be used as an aid in determining the scope of the claimed subject matter.

[0010a] According to one aspect of the present invention, there is provided a method performable by a media content processing system for enumerating media content stored on an MSD volume that is formatted using FAT, the method comprising the steps of: implementing in a memory a request queue for use by a file indexing process running on the media content processing system; reading a cluster containing directory or file data for the media content, the reading being performed so that clusters are read sequentially from the MSD volume, each sequentially read cluster having previously been identified as being associated with an item in the request queue; tracking a location of the cluster by (a) associating a request item with the cluster, (b) saving the request item in the request queue, (c) comparing the request item against an upper limit for a length of the request queue, the upper limit indicating that all clusters containing metadata associated with a given directory or file have been read, the metadata associated with an audio file and comprising two or more of album name, artist name, genre, track title, and track number, and (d) comparing the request item against a lower limit for the length of the request queue, the lower limit indicating that not all the clusters containing metadata associated with the given directory or file have been read; and iteratively performing the reading and tracking of clusters from the MSD volume until the upper limit is met and then parsing the directory or data to generate a file index for the media content, the metadata associated with the audio file used to create the file index.

[0010b] According to another aspect of the present invention, there is provided a non-transitory computer readable storage medium containing instructions which, when performed by one or more processors disposed in an electronic device, performs a method for creating a file

51331-868

index for data stored on an MSD volume, the method comprising the steps of: identifying an indexable file stored on the MSD volume by file extension; iteratively accessing clusters identified as being associated with the identified file in a sequential manner from the MSD volume using a plurality of passes through the MSD volume; examining a cluster to determine if

5 the cluster contains sufficient metadata to create an entry in the file index for the identified file, the metadata associated with an audio file and comprising two or more of album name, artist name, genre, track title, and track number; if the cluster contains sufficient metadata, then passing parseable data for the identified file from the cluster for inclusion in the file index; if the

10 cluster does not contain sufficient metadata, then storing a work item for the cluster in a queue in system memory of the device to indicate that one or more additional clusters are needed to enable passing parseable data for the identified file; and generating the file index using the parseable data, the metadata associated with the audio file used to create the file index.

[0010c] According to still another aspect of the present invention, there is provided a system for processing content stored on a volume that is formatted using FAT, comprising: a

15 file index processing layer supporting an indexing process for caching FAT table data, for queuing request items associated with portions of files or directories on the volume, the queued request items being used for caching data that is used for generating a file index for the processed content, and for finding request items for sequential reading from the volume, the file index for the processed content created using metadata stored as part of one or more audio files,

20 the metadata stored as part of the one or more audio files and comprising two or more of album name, artist name, genre, track title, and track number; a media core layer that is operatively coupled to the media processing layer and arranged for receiving call back notifications from the file index processing layer by which directory data or file data is identified to a caller operating in the media core layer; memory operatively coupled to the file indexing processing

25 layer for implementing a FAT table cache and implementing a request item queue arranged for queuing the request items; and an interface for reading content stored on the volume.

[0010d] According to yet another aspect of the present invention, there is provided a method performable by a media content processing system for enumerating a plurality of directories and files stored on a mass storage device, MSD, volume that is formatted using a file

51331-868

allocation table, FAT, file system, the method comprising the steps of: implementing in a memory a request queue for storing, by a file indexing process running on the media content processing system, request items associated with directories and files of the plurality of directories and files stored on the MSD volume; reading sequentially clusters from the MSD

5 volume, starting from the root directory, by iteratively: reading a cluster containing directory or file data for the media content; if the entire directory or file data has not been read, saving or updating a request item associated with the cluster in the request queue, otherwise removing from the request queue, if it exists, the request item associated with the cluster; as long as there are any request items in the request queue, if a number of request items in the request queue is

10 higher than a lower limit and lower than an upper limit, finding, in the request queue, the request item owning a next cluster closest to a current physical location of a read/write head of the MSD volume; if the request item owning the next closest cluster is associated to a file, the next closest cluster associated to said file is read and the method comes back to the step of determining if the entire directory or file data has been read; and if the request item owning the

15 next closest cluster is associated to a directory, the next subdirectory or file associated to said directory is read and the method comes back to the step of determining if the entire directory or file data has been read; and generating a file index for the media content using metadata contained in the clusters read for the plurality of directories and files of the MSD volume.

[0010e] According to a further aspect of the present invention, there is provided a non-

20 transitory computer readable storage medium containing instructions which, when carried out by one or more processors disposed in an electronic device, cause the one or more processors to perform the method as described herein.

[0010f] According to yet a further aspect of the present invention, there is provided a media content processing system for enumerating a plurality of directories and files stored on a

25 mass storage device, MSD, volume that is formatted using a file allocation table, FAT, file system, the media content processing system comprising means adapted to perform the method steps as described herein.

51331-868

DESCRIPTION OF THE DRAWINGS

[0011] FIG 1 is a simplified diagram showing an illustrative hard disk which uses low level formatting that is split into tracks, sectors, and clusters;

[0012] FIG 2 shows an illustrative sequence of cluster read operations in which clusters on a hard disk are accessed in random order;

[0013] FIG 3 shows an illustrative environment in which files and directories contained on a mass storage device are enumerated using a media content processing system that is located in a vehicle;

[0014] FIG 4 shows a layered architecture for the media content processing system shown in FIG 3;

[0015] FIG 5 is a flowchart for an illustrative method for processing file and directories that are contained on a mass storage device; and

[0016] FIG 6 shows an illustrative sequence of cluster read operations in which clusters on a hard disk are sequentially accessed.

[0017] Like reference numerals indicate like elements in the drawings.

DETAILED DESCRIPTION

[0018] FIG 1 is a simplified diagram showing an illustrative hard disk 100 which uses low level formatting that is split into tracks 105, sectors 112, and clusters 115 to support a FAT file system. The hard disk drives in mass storage devices ("MSDs") can use multiple hard disks (or "platters") that are arranged in stacked configuration. As shown in FIG 1, tracks 105 are configured in concentric circles and each track 105 comprises a number of sectors 112. A multiplicity of tracks 105 are used where the number is dependent on the size of the storage volume that is implemented using the hard disk 100. Each sector holds 512 bytes.

[0019] Clusters 115 comprise a set of sectors ranging in number from 2 to 128. The cluster size increases with the size of the hard disk 100 because FAT is limited in the number of clusters that it can track. Thus, larger volumes are supported in FAT by increasing the number of sectors per cluster. A cluster is the minimum space used by any read or write operation to the hard disk 100. Although clusters 115 are shown as being contiguous in FIG 1, the clusters associated with a given file or directory do not necessarily need to be contiguously located on the hard disk 100.

[0020] Various portions of the hard disk 100 are allocated for the FAT file system boot sector, one or more FAT tables, the root directory for volume, and a data region for files and directories. When a file is created, an entry is created in the FAT table and the first cluster number containing data is established. This entry in the FAT table either indicates that this is the last cluster of the file, or points to the next cluster. If the size of a file or directory is larger than the cluster size, then multiple clusters are allocated.

[0021] FIG 2 is a diagram which shows an illustrative sequence 200 of cluster read operations that occur when enumerating files and directories. In sequence 200, the clusters are accessed on the hard disk 100 (FIG 1) in random order using an existing FAT file enumeration methodology. In this example, several directories (named Dir1, Dir2 and Dir3), and several files are stored on the hard disk 100. The files are music files which are encoded in accordance with MP3 (Moving Picture Experts Group, MPEG-1, Audio Layer-3) which is a common standard for music. Dir1 includes File1.mp3 that is of sufficient size to span two clusters, and also includes File2.mp3 that is stored in a single cluster. The root directory includes File1.mp3 that is stored on disk in three clusters.

[0022] Because files and directories are written on the hard disk 100 to the first available clusters, the clusters storing such files and directories are accessed in a random manner as shown in FIG 2. When hard disk 100 is scanned using the FAT32 file system, for example, to enumerate its contents, the boot sector on the disk is consulted to locate the root directory indicated by numeral 210-1. The read/write head of the hard disk drive then moves to a location that is identified in the root directory to access Dir1, as indicated by reference numeral 210-2. The read/write head then goes to a location that is identified in Dir1 to access cluster 210-3 which is used to store the first cluster of music file File1.mp3.

[0023] To locate the next piece of the File1.mp3, the read/write head moves to consult the FAT table on the hard disk 100, and then moves to the identified cluster to access 210-4 as shown. The process of consulting the directory entries and/or the FAT table and then moving to the identified cluster repeats in order to access the remaining directories,

subdirectories, and files continues until all the contents on the hard drive are enumerated. Because the read/write head of the hard disk drive must continually move across the platters of the drive to get to the location of the FAT table, and to the clusters which store the files and directories, considerable latency may occur during enumeration of the volume's contents when using current FAT file system methodologies.

[0024] FIG 3 shows an illustrative environment 300 in which files and directories contained on an MSD 310 are enumerated using a media content processing system that employs the present direct MSD file indexing. In this example, environment 300 is an automotive environment in which a user employs the MSD 310 to store media content including music that the user desires to be rendered (i.e., played) over a sound/entertainment system 316 and speakers 319 that are located in a vehicle 321. However, it is emphasized that the environment 300 is merely illustrative and that the present direct MSD file indexing is not limited to automotive applications or music files. It is further contemplated that the benefits of direct MSD file indexing may be applied to any type of content (for example, data and other media content such as photographs and video) that is stored on an MSD using the FAT file system in a variety of different applications and implementations.

[0025] MSD 310, in this example, is a conventional hard disk-based device that is configured to be compact and portable and is further arranged as a volume under the FAT32 file system. MSD 310 is coupled to the sound/entertainment system 316 in the vehicle 321 using a USB cable 325 that carries signals in compliance with USB 2.0, although in alternative implementations other data transfer busses and protocols may also be utilized, including those, for example which use wireless or optical infrastructure.

[0026] A media content processing system 332 is also operative in the environment 300. In this example, media content processing system 332 is a discrete system in the vehicle 321 and is typically located behind the dashboard or console area, although other locations may also be utilized as dictated by the circumstances of a particular implementation. The media content processing system 332 is configured to be operatively connectable to the sound/entertainment system 316 over an interface (not shown), or it may be optionally integrated with the functionality provided by the sound/entertainment system 316 in common package or form factor in some applications. Media content processing system 332 is shown in detail in FIG 4 and described in the accompanying text below.

[0027] As shown in FIG 4, the media processing system 332 includes a layered architecture that comprises a media player 406, a media core 411, and a file index

processing layer 415. The media player 406 is arranged to provide user interface (“UI”) functionality by exposing, in this illustrative example, a file index for data, including media content such as music, which is stored on the MSD 310. Thus, for example, when a user plugs the MSD 310 into the sound/entertainment system 316, the media player 406 functions to provide enumeration of the music on the MSD 310 in an indexed list that is displayed on a screen or other UI device from which the user may browse and select items to be played.

[0028] The media core 411 is arranged to parse file and/or directory data received from a process operating in the file index processing layer 415 to thereby perform the file enumeration through call back and return messages, as respectively indicated by reference numerals 418 and 422. Media core 411 may be optionally arranged to provide additional features and functionalities including, for example, media content decoding, rendering, and playback control in some implementations.

[0029] The file index processing layer 415 includes a direct MSD file indexing process 430 which interacts with the media core 411, as shown, and which also interacts with a FAT table cache 432 and a request queue 435. The direct MSD file indexing process 430 is further configured to read data from the MSD 310 that is sent using the USB protocol, in this illustrative example, as indicated by reference numeral 437.

[0030] The FAT table cache 432 is used to cache FAT table data whenever it is read from the hard disk 100 (FIG 1). This caching is performed due to the likelihood that the next required FAT table lookup for data of interest will be included in any recently read FAT table data. Caching such data may reduce the necessity of the read/write head having to move back to consult the FAT table on the hard disk which can advantageously reduce the latency in file enumeration.

[0031] The FAT table cache 432 and request queue 435 are implemented in system memory 439 (e.g., volatile random access memory or “RAM”). The interaction between the FAT table cache 432 and direct MSD file indexing process 430 includes caching FAT table data, as indicated by reference numeral 440, and reading FAT table data from the cache, as indicated by reference numeral 442. The interaction between the request queue 435 and direct MSD file indexing process 430 includes saving request items in the queue, as indicated by reference numeral 445, and reading request items from the queue, as indicated by reference numeral 448. The operation of the direct MSD file indexing process 430 is shown in the flowchart in FIG 5 and described in the accompanying text.

[0032] FIG 5 is a flowchart for an illustrative method 500 performed by the media content processing system 332 for processing files and directories that are contained on the mass storage device 310. The method starts at block 505 at the root directory. At block 512, an entry is read in a directory (e.g., either the root directory or a directory on the hard disk 100) to identify a file or subdirectory.

[0033] At block 516, the direct MSD file indexing process 430 notifies the caller (i.e., the media core 411) of the new data ascertained from the method step at block 512. Control passes to decision block 520 where the caller decides whether it is interested in the new data. For example, the file extension may be of a particular type that is utilized in the illustrative environment 300 such as an MP3, WMA (Windows® Media Audio), or WAV (WAVEform audio format) file. In this case then, data associated with non-audio formats or file extensions would not be of interest.

[0034] Another example for which the caller may not be interested in the data is where enough parts of file have already been located so as to identify particular metadata of interest that will be used to enumerate the stored content and create a file index. Typically, and in this illustrative example, the metadata of interest relates to music and includes album name, artist name, genre, track (e.g., song) title, track number, etc. Thus, if all the metadata is already located, then the caller will not need to continue with an item even when it is a logical part of a file that was previously identified as being of interest. While such logical parts of the file would be needed to play back the content, they are not needed for enumeration purposes and could thus be skipped.

[0035] If the data is of interest to the caller, then control passes to decision block 523 where the direct MSD file indexing process 430 determines if the entire directory or file has been read. If it has not, then an item is either saved or updated in the request queue 435, as indicated at block 526. If the data is not of interest to the caller, then control passes to block 530, and an item is either not added, or removed, from the request queue.

[0036] Control passes from either block 526 or block 530 to decision block 534 where the direct MSD file index process 430 determines if there are any items in the request queue 435. If so, then control passes to decision block 538 where the direct MSD file indexing process 430 determines if the number of items in the request queue 435 is less than a low water mark (i.e., a lower limit). If so, then at decision block 542, if there are any directory items in the request queue 435, control returns to block 512 where the next sub-directory or file associated with that directory item in the request queue 435 is read. The

low water mark is used to designate a set minimum number of items in the request queue 435 above which it is efficient to process the queued items.

[0037] If there are no directory items in the request queue 435, then control passes to block 545 where the next data cluster that is associated with that file item in the request queue 435 is read.

[0038] If the number of items is not below the low water mark, then control passes to block 547. If the number of items in the request queue 435 is greater than a high water mark (i.e., an upper limit), then control passes to block 550. If there are no file items in the request queue 435, then control returns to block 512 where the next sub-directory or file associated with that directory item in the request queue 435 is read.

[0039] If there are file items in the request queue 435, then control passes to block 545 where the next data cluster that is associated with that file item in the request queue 435 is read. If the number of items in the request queue 435 is less than the high water mark, then control passes to block 552 where the file item in the request queue 435 that owns the next closest cluster is found. At decision block 554, if the item in the request queue 435 is a file, then control passes to block 545 where the next data cluster that is associated with that file item in the request queue 435 is read. If the next item is not a file (i.e., it is a directory), then control returns to block 512 where the next sub-directory or file associated with that directory item in the request queue 435 is read. The high water mark may be configured to different values depending on the requirements of a particular implementation and will typically be sized in light of available resources such as system memory.

[0040] The above described method is successively iterated until, at block 534, when there are no more items remaining in request queue 435, the method ends at block 560.

[0041] FIG 6 shows an illustrative sequence 600 of cluster read operations in which clusters on the hard disk 100 (FIG 1) are accessed in sequence using the method shown in FIG 5 and described in the accompanying text. The clusters are associated with the same directories and files as shown in FIG 2. As shown in FIG 6, the clusters are read sequentially to minimize read/write head movement on the hard disk 100 which advantageously reduces latency in file indexing.

Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

51331-868

CLAIMS:

1. A method performable by a media content processing system for enumerating media content stored on an MSD volume that is formatted using FAT, the method comprising the steps of:

5 implementing in a memory a request queue for use by a file indexing process running on the media content processing system;

reading a cluster containing directory or file data for the media content, the reading being performed so that clusters are read sequentially from the MSD volume, each sequentially read cluster having previously been identified as being associated with an item in
10 the request queue;

tracking a location of the cluster by (a) associating a request item with the cluster, (b) saving the request item in the request queue, (c) comparing the request item against an upper limit for a length of the request queue, the upper limit indicating that all clusters containing metadata associated with a given directory or file have been read, the
15 metadata associated with an audio file and comprising two or more of album name, artist name, genre, track title, and track number, and (d) comparing the request item against a lower limit for the length of the request queue, the lower limit indicating that not all the clusters containing metadata associated with the given directory or file have been read; and

iteratively performing the reading and tracking of clusters from the MSD
20 volume until the upper limit is met and then parsing the directory or data to generate a file index for the media content, the metadata associated with the audio file used to create the file index.

2. The method of claim 1 including the steps of caching a FAT table in the memory and reading data from the cached FAT table.

25 3. A non-transitory computer readable storage medium containing instructions which, when performed by one or more processors disposed in an electronic device, performs

51331-868

a method for creating a file index for data stored on an MSD volume, the method comprising the steps of:

identifying an indexable file stored on the MSD volume by file extension;

5 iteratively accessing clusters identified as being associated with the identified file in a sequential manner from the MSD volume using a plurality of passes through the MSD volume;

examining a cluster to determine if the cluster contains sufficient metadata to create an entry in the file index for the identified file, the metadata associated with an audio file and comprising two or more of album name, artist name, genre, track title, and track
10 number;

if the cluster contains sufficient metadata, then passing parseable data for the identified file from the cluster for inclusion in the file index;

if the cluster does not contain sufficient metadata, then storing a work item for the cluster in a queue in system memory of the device to indicate that one or more additional
15 clusters are needed to enable passing parseable data for the identified file; and

generating the file index using the parseable data, the metadata associated with the audio file used to create the file index.

4. The computer-readable medium of claim 3 in which the file extension is associated with an audio file.

20 5. A system for processing content stored on a volume that is formatted using FAT, comprising:

a file index processing layer supporting an indexing process for caching FAT table data, for queuing request items associated with portions of files or directories on the volume, the queued request items being used for caching data that is used for generating a file
25 index for the processed content, and for finding request items for sequential reading from the

51331-868

volume, the file index for the processed content created using metadata stored as part of one or more audio files, the metadata stored as part of the one or more audio files and comprising two or more of album name, artist name, genre, track title, and track number;

5 a media core layer that is operatively coupled to the media processing layer and arranged for receiving call back notifications from the file index processing layer by which directory data or file data is identified to a caller operating in the media core layer;

memory operatively coupled to the file indexing processing layer for implementing a FAT table cache and implementing a request item queue arranged for queuing the request items; and

10 an interface for reading content stored on the volume.

6. The system of claim 5 in which the interface comprises a high speed data interface selected from one of USB or IEEE-1394.

7. The system of claim 5 in which the interface uses communication infrastructure selected from one of wireless infrastructure or optical infrastructure.

15 8. The system of claim 5 in which the media core layer is further arranged to render the content read from the volume.

9. The system of claim 5 further including a media player layer that is arranged to expose a user interface by which the file index is displayable to an end-user.

10. The system of claim 9 in which the media player layer is further arranged to
20 expose a user interface by which items in the file index are selectable by the end-user.

11. The system of claim 5 further including functionalities for audio and video processing attendant to provisioning of an entertainment subsystem usable in a vehicle environment.

12. The system of claim 5 in which the volume is implemented in an MSD.

51331-868

13. The system of claim 12 in which the MSD is a hard disk drive-type MSD.
14. The system of claim 5 in which the process for queuing request items is performed iteratively until all the content is read from the volume.
15. The system of claim 5 in which the process for queuing request items is performed in a manner to maintain a queue length between an upper limit and a lower limit.
16. The system of claim 5 in which the content stored on the volume is media content.
17. The system of claim 16 in which processing of a file is terminated upon location of predetermined metadata associated with the media content.
18. The system of claim 17 in which the metadata is selected from one of track title, track number, artist name, album name, or genre.
19. The system of claim 5 in which the portions of files or directories on the volume are stored on a cluster basis on the volume.
20. A method performable by a media content processing system for enumerating a plurality of directories and files stored on a mass storage device, MSD, volume that is formatted using a file allocation table, FAT, file system, the method comprising the steps of:
 - implementing in a memory a request queue for storing, by a file indexing process running on the media content processing system, request items associated with directories and files of the plurality of directories and files stored on the MSD volume;
 - reading sequentially clusters from the MSD volume, starting from the root directory, by iteratively:
 - reading a cluster containing directory or file data for the media content;

51331-868

if the entire directory or file data has not been read, saving or updating a request item associated with the cluster in the request queue, otherwise removing from the request queue, if it exists, the request item associated with the cluster;

5 as long as there are any request items in the request queue, if a number of request items in the request queue is higher than a lower limit and lower than an upper limit,

finding, in the request queue, the request item owning a next cluster closest to a current physical location of a read/write head of the MSD volume;

10 if the request item owning the next closest cluster is associated to a file, the next closest cluster associated to said file is read and the method comes back to the step of determining if the entire directory or file data has been read; and

if the request item owning the next closest cluster is associated to a directory, the next subdirectory or file associated to said directory is read and the method comes back to the step of determining if the entire directory or file data has been read; and

15 generating a file index for the media content using metadata contained in the clusters read for the plurality of directories and files of the MSD volume.

21. The method of claim 20 including the steps of caching a FAT table in the memory and reading data from the cached FAT table.

22. The method of claim 20 in which at least one file of the MSD volume is an audio file, the metadata of which comprises one of album name, artist name, genre, track title,
20 or track number.

23. The method of one of claims 20 to 22, further comprising, after each reading step in the iteration:

examining a cluster to determine if the cluster contain sufficient metadata to create an entry in the file index for the identified file;

51331-868

if the cluster contains sufficient metadata, then passing the metadata for the identified file from the cluster for inclusion in the file index and removing from the request queue, if it exists, the request item associated with the cluster; and

- 5 if the cluster does not contain sufficient metadata, then going to step of determining whether the entire directory or file data has been read to indicate that one or more additional clusters are needed to enable passing megadata for the identified file.

24. A non-transitory computer readable storage medium containing instructions which, when carried out by one or more processors disposed in an electronic device, cause the one or more processors to perform the method of one of claims 20 to 23.

- 10 25. A media content processing system for enumerating a plurality of directories and files stored on a mass storage device, MSD, volume that is formatted using a file allocation table, FAT, file system, the media content processing system comprising means adapted to perform the method steps of one of claims 20 to 23.

1/5

FIG. 1

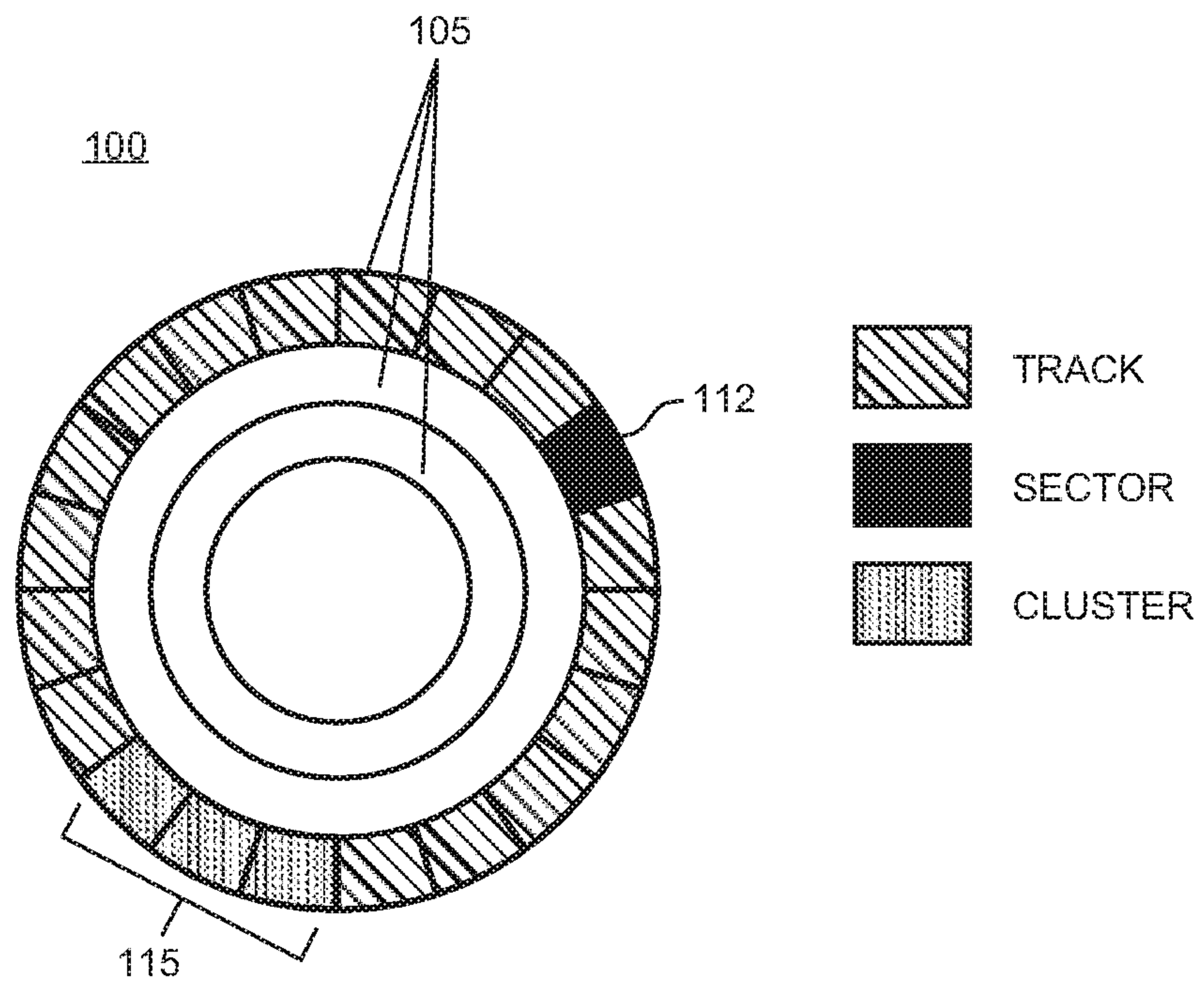
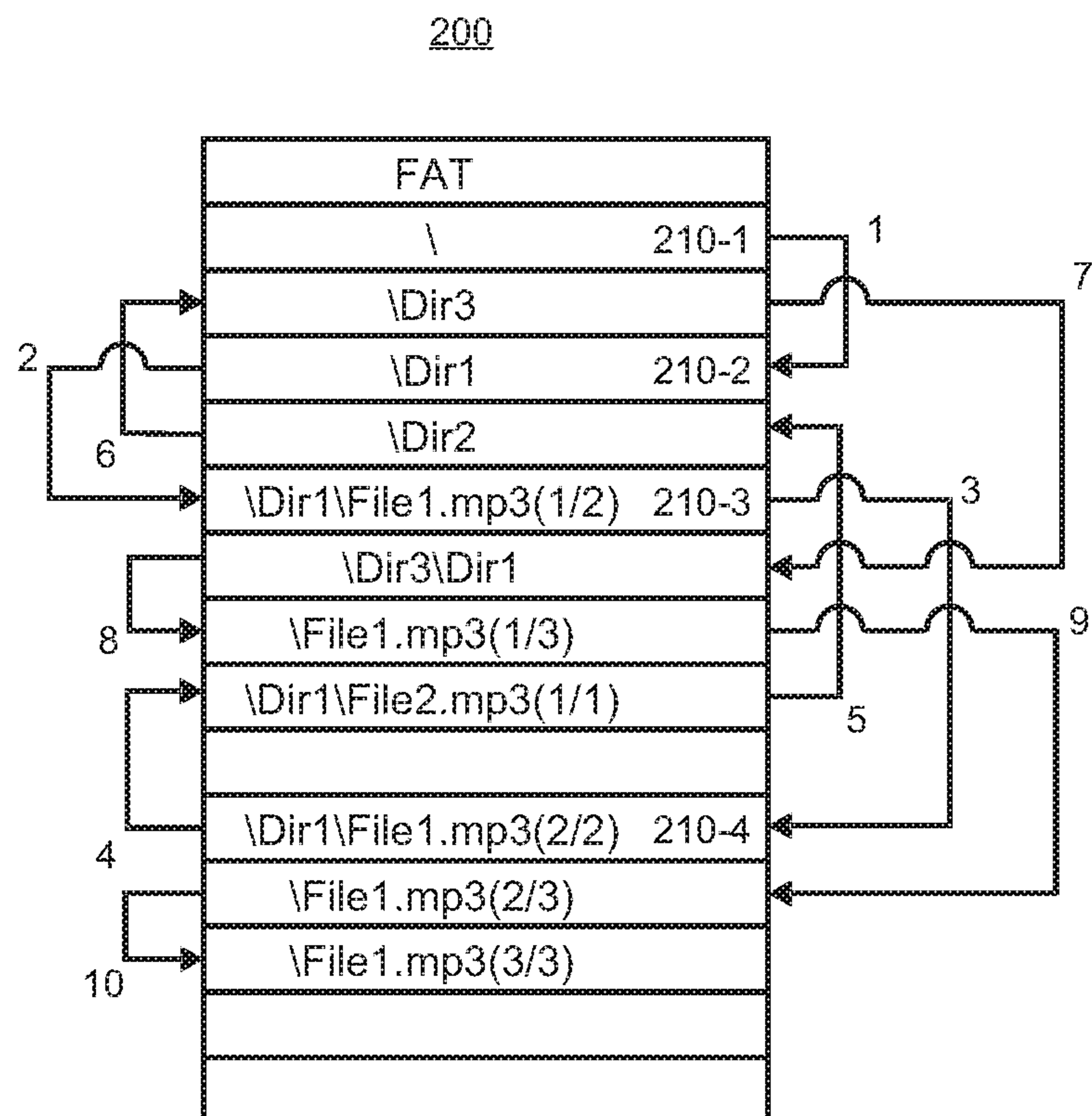
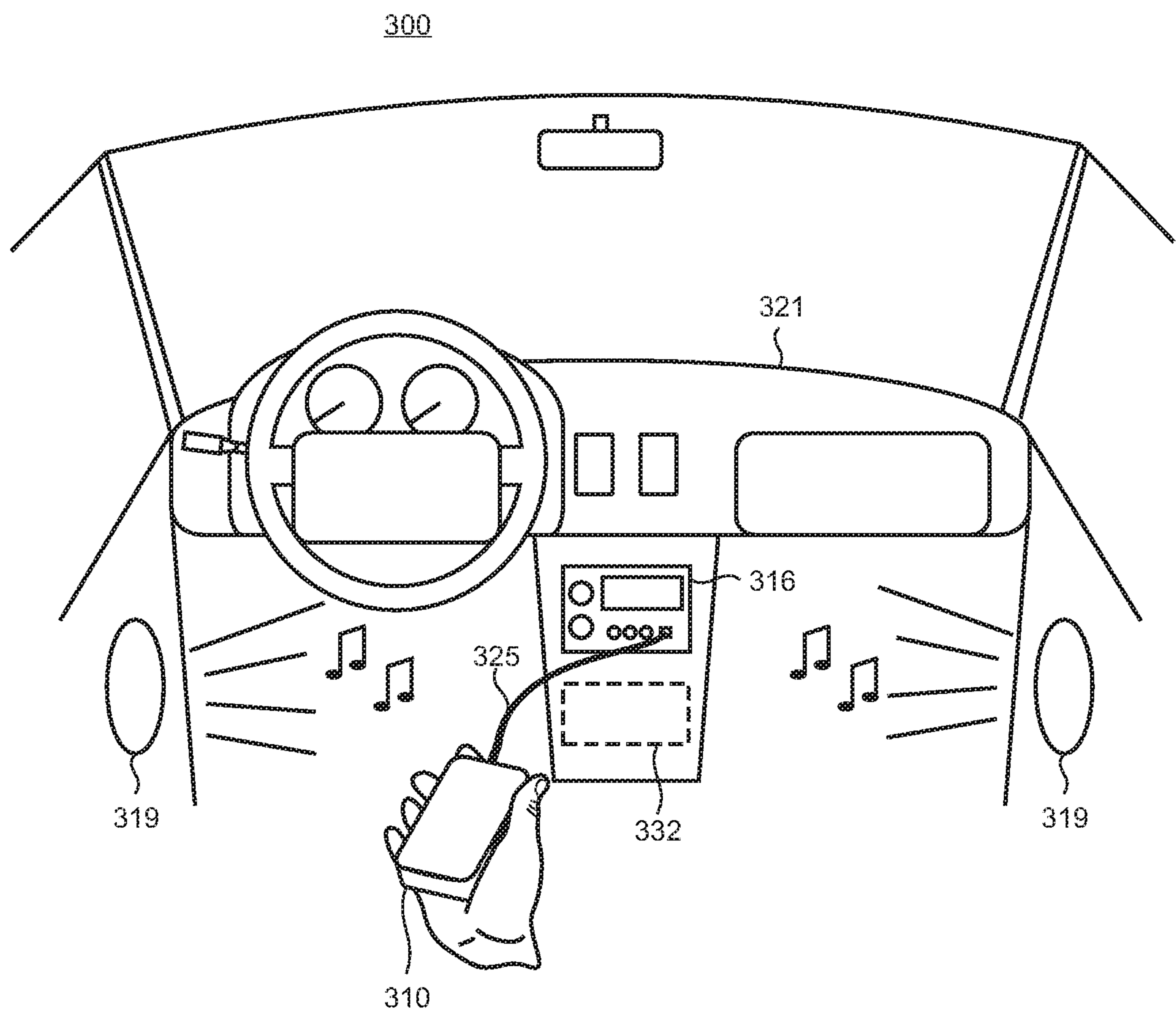


FIG. 2



2/5

FIG. 3



3/5

FIG. 4

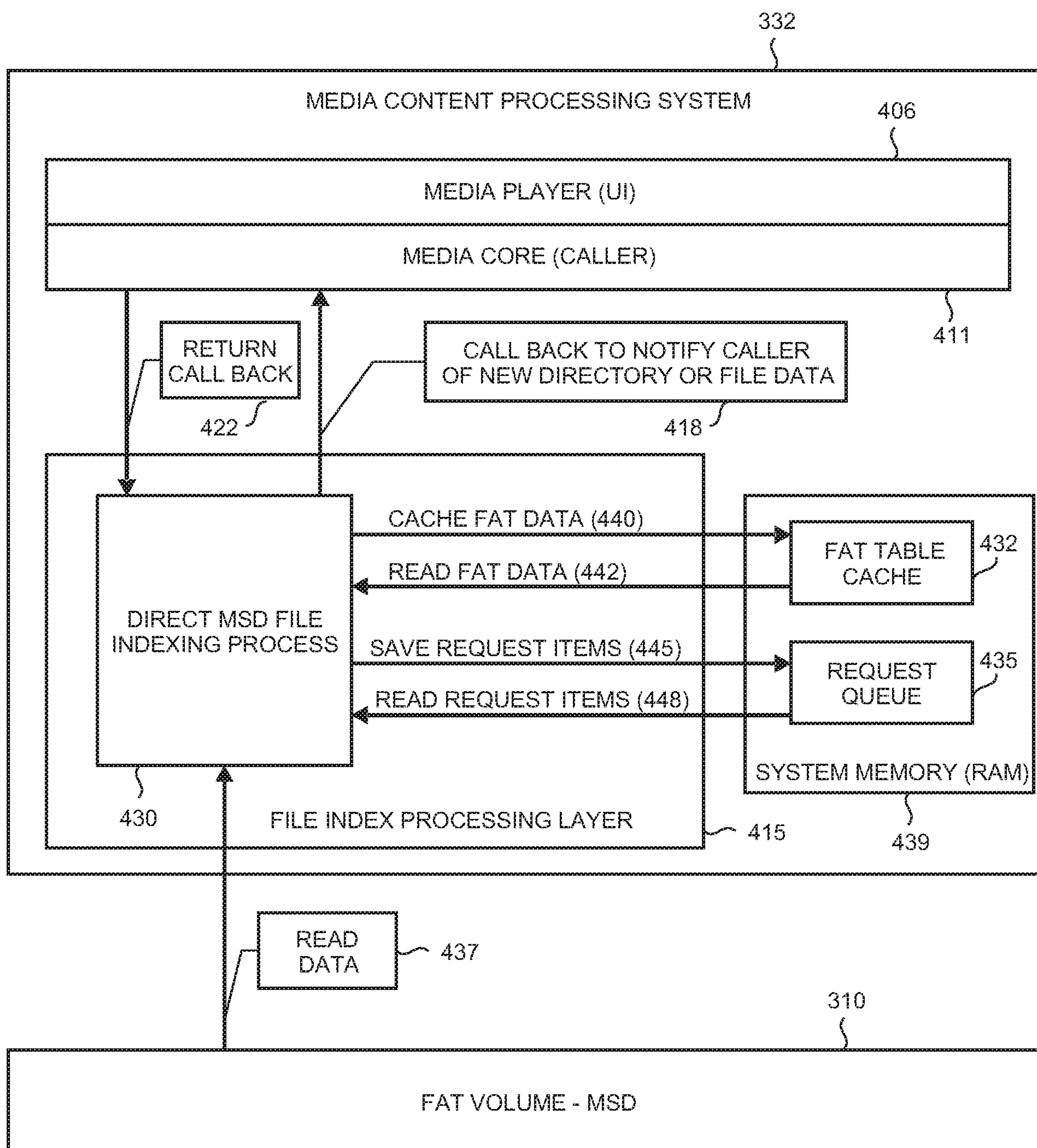
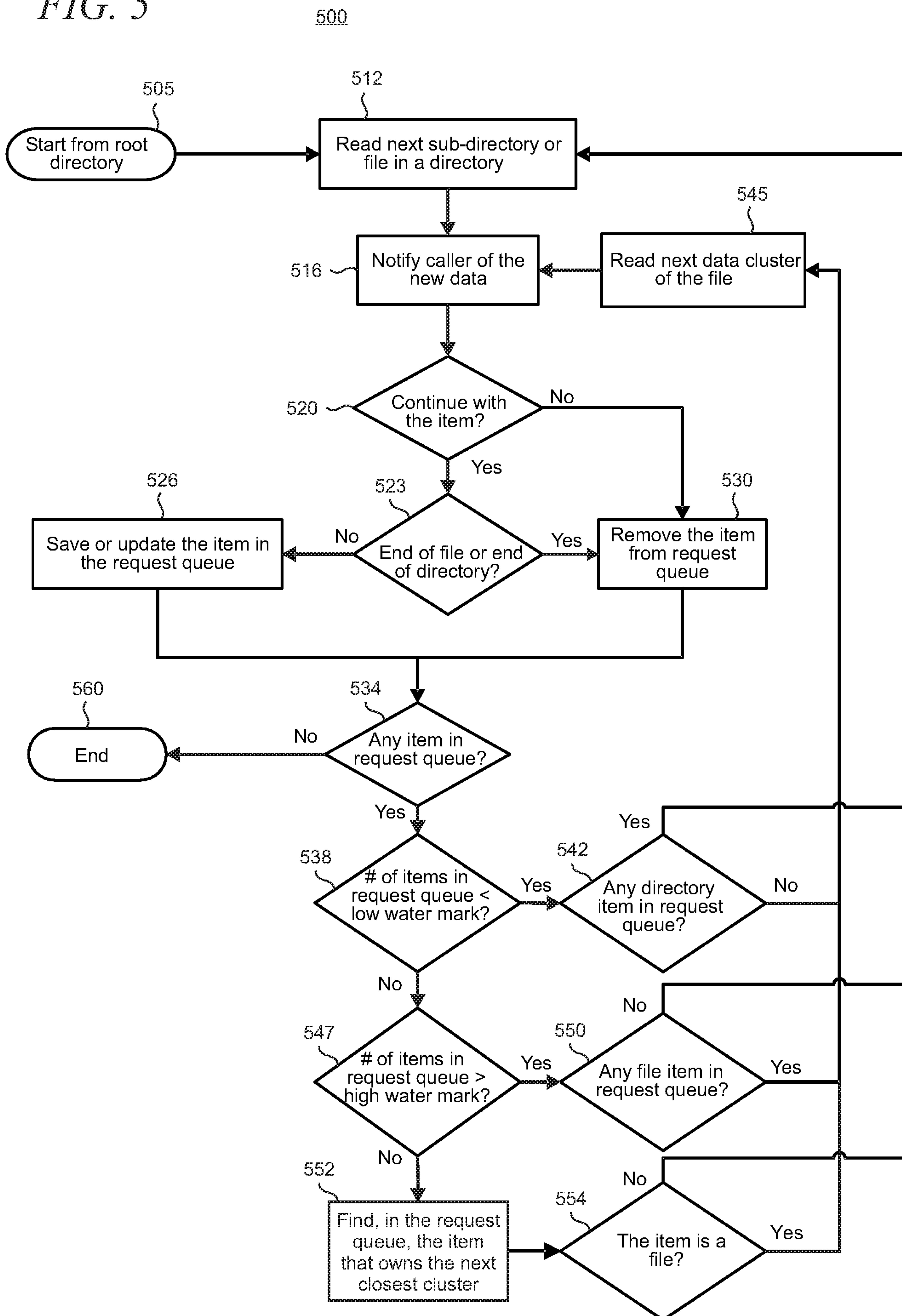


FIG. 5

4/5



5/5

FIG. 6

