(12) UK Patent (19) GB (11) 2500237 (13) B

(45) Date of B Publication 23.09.2020

(54) Title of the Invention: Data storage system

(51) INT CL: ***G06F 16/182*** (2019.01)

(21) Application No: **1204560.5**

(22) Date of Filing: **15.03.2012**

(43) Date of A Publication **18.09.2013**

(56) Documents Cited:

US 6742020 B1 **US 20080077635 A1**
US 20040243575 A1 **US 20040139222 A1**
"Information Sharing in Mobile Ad-Hoc Networks: Metadata Management in the MIDAS Dataspace" Munthe-Kaas, E et al. Mobile Data Management: Systems, Services and Middleware 2009. IEEE computer society pages 252 to 259

(58) Field of Search:

As for published application 2500237 A viz:
INT CL **G06F**
Other: **WPI, EPODOC, TXTEN,TXTT, XPI3E**
updated as appropriate

Additional Fields
INT CL **H04L**
Other: **None**

(72) Inventor(s):

Julian Chesterfield

(73) Proprietor(s):

OnApp Limited
The Cooperage, Old Truman Brewery, 91 Bricklane, LONDON, E1 6QL, United Kingdom

(74) Agent and/or Address for Service:

Gill Jennings & Every LLP
The Broadgate Tower, 20 Primrose Street, LONDON, EC2A 2ES, United Kingdom

GB 2500237 B

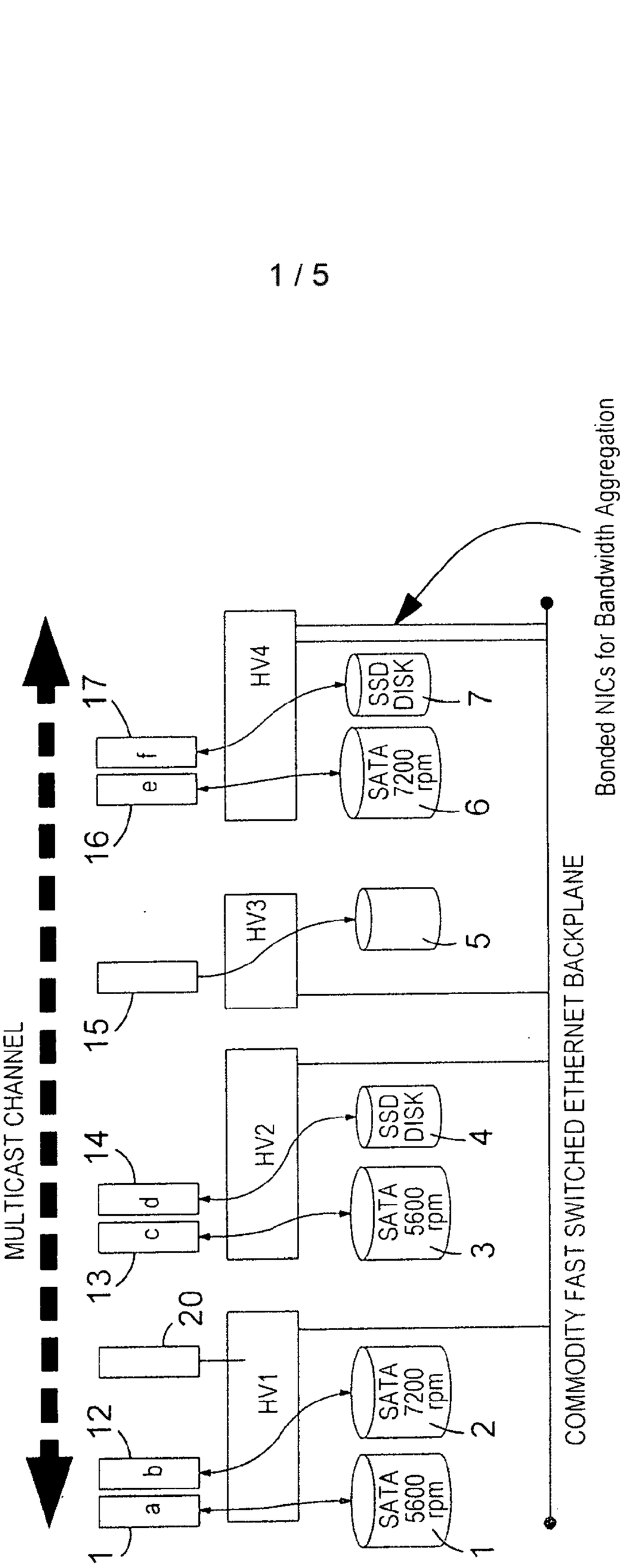
MULTICAST CHANNEL
COMMODITY FAST SWITCHED ETHERNET BACKPLANE
Bonded NICs for Bandwidth Aggregation
HV1
HV2
HV3
HV4
SATA 5600 rpm
SATA 7200 rpm
SSD DISK
1
2
3
4
5
6
7
a
b
c
d
e
f
12
13
14
15
16
17
20

FIG.1

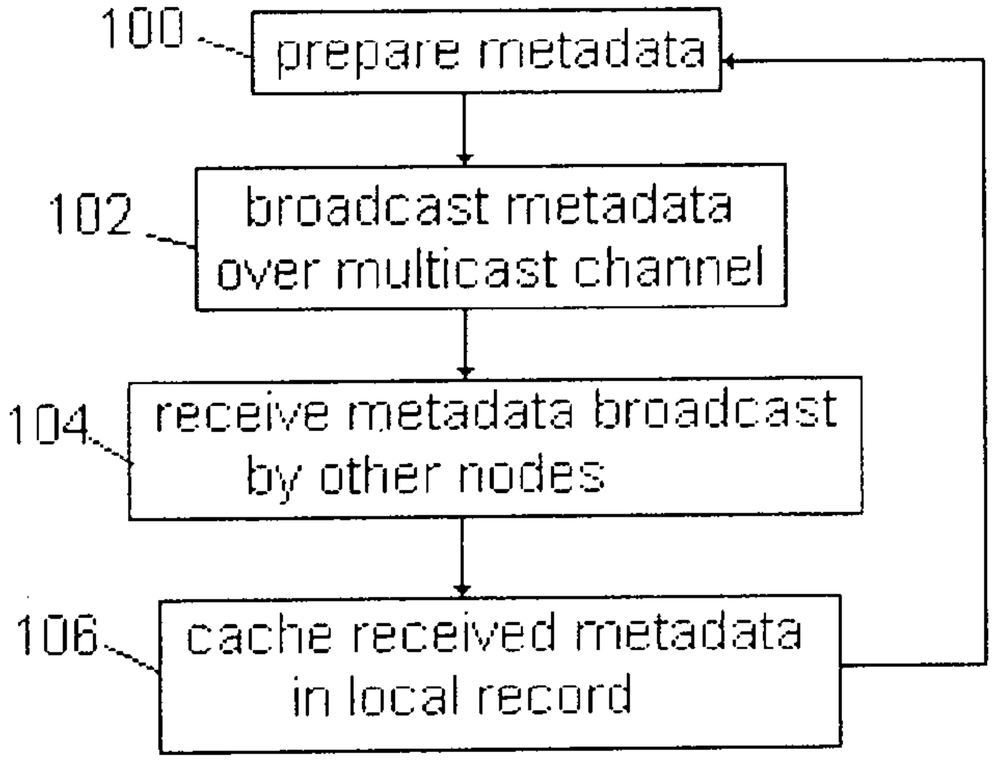
100
prepare metadata
102
broadcast metadata over multicast channel
104
receive metadata broadcast by other nodes
106
cache received metadata in local record

Fig. 2

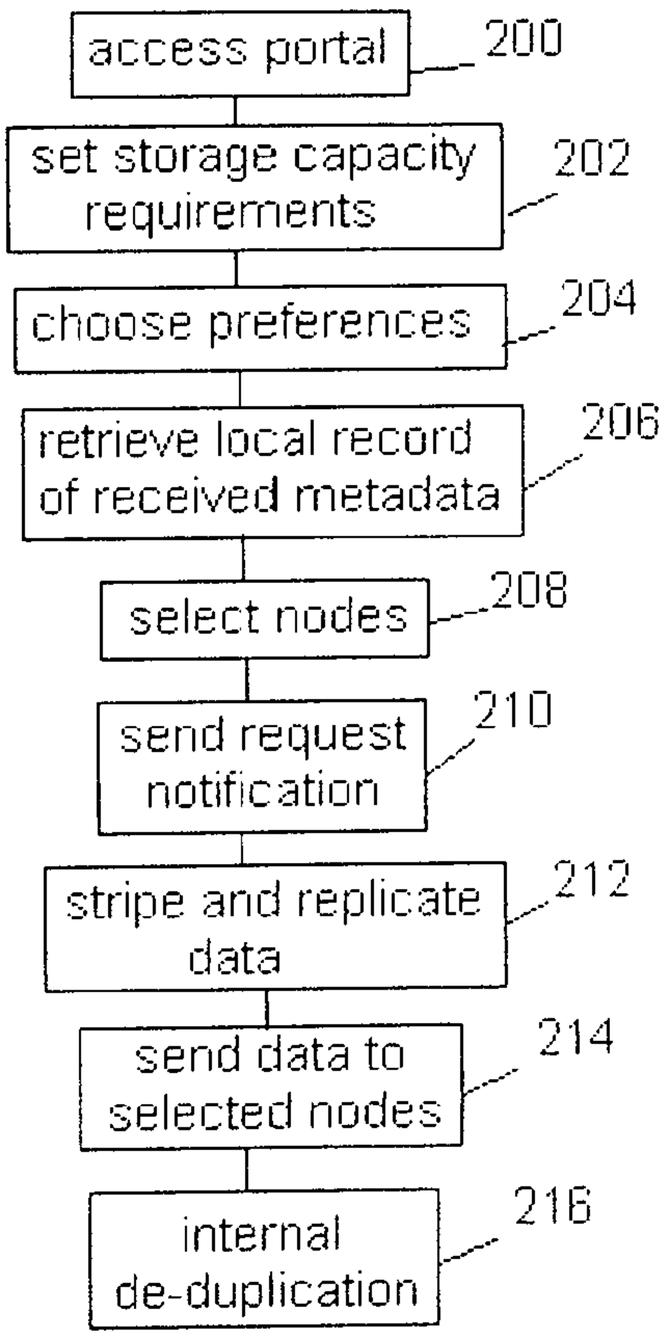
access portal
200
set storage capacity requirements
202
choose preferences
204
retrieve local record of received metadata
206
select nodes
208
send request notification
210
stripe and replicate data
212
send data to selected nodes
214
internal de-duplication
216

Fig. 3

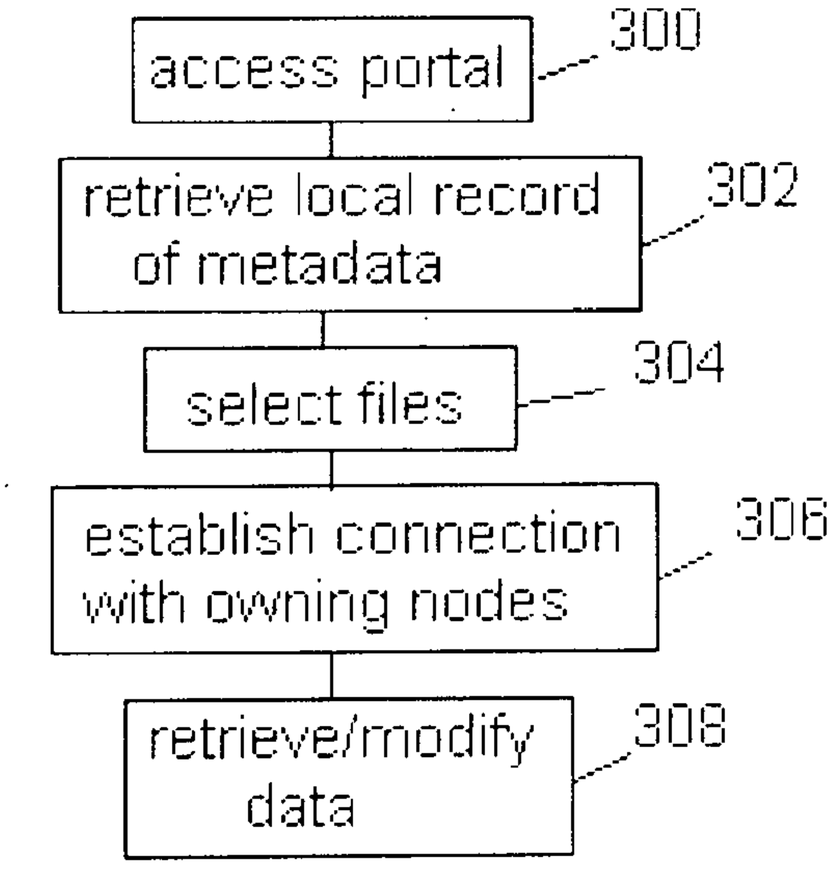
access portal
300
retrieve local record of metadata
302
select files
304
establish connection with owning nodes
306
retrieve/modify data
308

Fig. 4

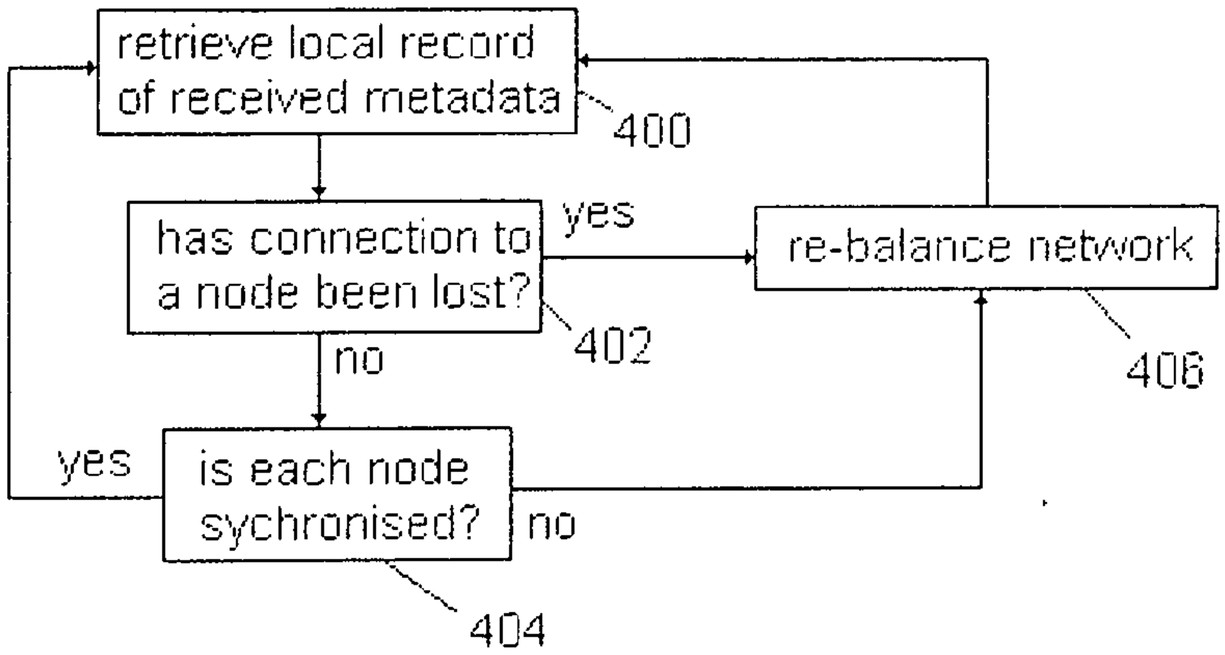
retrieve local record of received metadata
400
has connection to a node been lost?
yes
402
re-balance network
406
no
yes
is each node sychronised?
no
404

Fig. 5

Data Storage System

The present invention relates to a data storage system in which multiple data storage units are connected together over a network.

Cloud storage is an important development in computing whereby users can upload data through an internet portal so that the data can be stored remotely and accessed later from any location or computer. Typically, cloud storage providers maintain data centres for the remote storage of large amounts of data. A data centre typically includes an array of data storage disks, controlled by a central controller. The central controller manages read/write operations to the disks as well as all input/output processes to and from clients that communicate over the internet. A client is likely to be an internet portal that can translate an input/output request from a data centre into something that an end user can handle, such as a file object or a web-based API (Application Programming Interface) call.

A group consisting of a data centre and one or more clients that communicate over a network is sometimes referred to as a Storage Area Network (SAN). A network in this context might be SCSI over IP, SCSI over fibre channel, SCSI over Ethernet, ATA over Ethernet, or others.

The storage capacity of a data centre is generally limited by the properties of the central controller. The central controller can only cope with a finite number of processes at any time, and this places an effective upper limit on the number of data storage disks that can be present, and therefore the overall storage capacity of the system.

According to an aspect of the present invention there is provided a computer data storage network, comprising:

a plurality of storage nodes connected together to provide an integrated storage resource;

wherein each storage node is configured to broadcast metadata across the network concerning its stored data, and each storage node is configured to receive metadata that are broadcast from other storage nodes; and

wherein each storage node is configured to maintain a local record of the metadata received over the network; wherein each storage node is configured to broadcast metadata to the network repeatedly, with a predetermined frequency; and wherein the frequency is selected in dependence on the number of storage nodes in the network according to the local record of the received metadata.

By maintaining a local record of the metadata received over the network, each storage node can establish a live picture of the data stored in the other storage nodes. This can permit any individual storage node to take decisions that control the distribution of data in the network. In this way the network can be organised without any central controller because this is replaced by the co-operative decisions of the distributed storage nodes.

The data storage capacity in the integrated resource can be increased by adding further storage nodes, without limit. The system can be scaled easily because there is no central controller whose resources will be depleted by additional nodes. Each additional storage node provides its own processing capabilities, including its own input/output queue, without placing any demands on a common resource. In this way the data storage capacity in the network can be increased by a factor of 10 or 100 without any difficulty.

In this decentralised system, no individual storage node is critical to the network. This means that there is no single point of management or potential failure in the network.

Preferably each storage node is configured to control its own read/write operations as well as communications with other storage nodes in the network. In this way, a cloud Storage Area Network (SAN) can be provided where a number of remote storage nodes are linked together to become an integrated storage resource. The network is decentralised because each storage node can control its own read/write operations and communications over the network. Thus, there is no central controller that provides overall control of the integrated storage resource.

Preferably each storage node comprises a storage unit in which data can be stored, and a software controller for controlling read/write operations, communications with other storage nodes, and the broadcast metadata.

The storage unit may be any type of storage medium, such as a solid-state drive (SSD) or a hard-disk drive (HDD). A number of different types of storage media are typically

17 01 20

provided across the network to provide different performance requirements according to user preferences.

The software controller is an embedded Operating System (OS) controller that manages content stored in the physical storage unit. The software controller preferably runs in a virtual machine to enable resource isolation, using the existing hardware in the storage node. The software controller preferably accesses the storage unit directly from the control domain.

Preferably one or more of the storage nodes includes an interface through which a user can upload, modify and/or access data. Any storage node can include a user interface. In practice, however, only a subset of the storage nodes provide the facility for users to upload, modify and access stored data. The interface is preferably provided from the control domain of a hypervisor or from a storage node running as a virtual machine.

A storage node may be configured to select a number of storage nodes when new data are uploaded via the interface, wherein the selection is made using the local record of metadata received over the network. In this way the storage node can select a number of “owning” nodes for a particular set of data. The storage node can then distribute the new data across the owning nodes so that a desirable redundancy is achieved and so that the data can be accessed at a particular rate. Each owning node receives a read or write request from the requesting storage node and executes the storage of written data under its own control and discretion.

The selection of storage nodes is made using the local record of metadata received over the network. The selection criteria may vary according to user preferences. In one example, storage nodes may be selected according to their available data storage capacity.

A user would need to access each of the selected storage nodes in order to edit the uploaded data. A user can be directed to all of the relevant storage nodes easily because all of the storage nodes are aware of the owning nodes for a particular set of data through the metadata that they receive.

The storage nodes may be selected according to at least one of physical location, available data storage capacity, redundancy properties, maximum bandwidth, and disk speed/performance. In one arrangement the storage nodes may be selected to maximise the speed at which data can be accessed. Thus, storage nodes may be selected that have a high maximum bandwidth and that are geographically nearby a user.

In another arrangement storage nodes may be selected according to user preferences. A user may specify that they would prefer their data to be stored with high redundancy or so that it can be accessed rapidly. Storage nodes that offer these properties may be selected accordingly.

In certain arrangements the storage nodes can be selected according to the prices charged for data storage in particular storage nodes. Thus, the storage nodes may be selected so that the data storage costs are minimised, even if this sacrifices performance. Storage node selection based on price is not critical to the key functionality.

The storage node that performs the selection may also be configured to upload the new data to the selected storage nodes. In addition the storage node may be configured to stripe and replicate the new data before it is uploaded to the selected storage nodes. Striping data is desirable as it increases the rate at which the data can be accessed. Replicating the data is desirable as it creates redundancy in the new data.

An owning node preferably stores a single copy of data, referenced by a stripe number. Any single owning node will preferably have a single copy of one of the logical data stripes.

Each storage node is configured to broadcast metadata to the network repeatedly, with a predetermined frequency. By broadcasting metadata on a regular basis each storage node can advertise its current state so that the other storage nodes retain an accurate picture of the state of the network in their local record of received metadata. In addition, each storage node may be configured to detect a possible error if metadata are not received from a particular storage node within a predetermined period of time.

The broadcast frequency is selected in dependence on the number of storage nodes in the network according to the local record of the received metadata. More specifically the broadcast frequency is preferably inversely proportional to the number of storage nodes in the network. This is advantageous so that the bandwidth required by broadcast metadata does not increase exponentially when the network expands to include new storage nodes. Preferably the broadcast frequency is selected so that the bandwidth required by broadcast metadata remains substantially constant as the number of nodes changes.

Each storage node may comprise a re-balancing module that is configured to re-balance data in the network when predetermined criteria are satisfied. Re-balancing data may involve a re-distribution of data. This may be necessary, for example, if connection to the one of the storage nodes is lost. In these circumstances a re-balancing module in one of the storage nodes may expel the relevant storage node from a group of 'owning' nodes for a particular data set, and a new storage node may be added. The data that were stored in the expelled storage node may be copied to the new storage node, using redundant data in the network. The broadcast metadata from each storage node may reflect the fact that a storage node has been expelled and a new storage node has been added.

Re-balancing may also occur if one of the storage nodes in the network is 'out of synch' with the other nodes. Each storage node may include an entry in its broadcast metadata indicating its time of last update. Preferably all of the storage nodes are updated synchronously so that, together, they contain an up-to-date data set. A storage node may contain old data if it has temporarily lost contact with one or more of the storage nodes.

The metadata broadcast by each storage node may include information concerning the properties of the node. In addition, each storage node may include a de-duplication module that is configured to remove duplicate entries in its stored data.

According to an aspect of the present invention there is provided a method of operating a computer data storage network that comprises a plurality of storage nodes connected together to provide an integrated storage resource, the method comprising the steps of:

providing independent control logic at each storage node for read/write operations as well as communications with other storage nodes in the network;

broadcasting metadata across the network from each storage node concerning the data stored therein;

receiving the broadcast metadata at each storage node;

maintaining a record at each storage node concerning the metadata received over the network;

broadcasting the metadata to the network repeatedly, with a predetermined frequency; and

selecting the frequency in dependence on the number of storage nodes in the network according to the local record of the received metadata.

Any apparatus features may be embodied as method steps and *vice-versa*.

According to another aspect of the invention a computer readable storage medium is provided having a computer program stored thereon, the computer program comprising:

a program module configured to provide control logic at a storage node for read/write operations and control logic for communicating with other storage nodes in a network;

a program module configured to control a storage node to broadcast metadata across the network concerning the data stored therein;

a program module configured to receive the broadcast metadata from other storage nodes in the network; and

a program module configured to maintain a record concerning the metadata received over the network;

wherein the program module configured to broadcast metadata is configured to broadcast metadata to the network repeatedly, with a

predetermined frequency; and

wherein the frequency is selected in dependence on the number of storage nodes in the network according to the local record of the received metadata.

Preferred features of the present invention will now be described, purely by way of example, with reference to the accompanying drawings, in which:

Figure 1 is a representation of a computer storage network in an embodiment of the invention;

Figure 2 is a flow chart showing a sequence of steps undertaken by a storage node in a computer storage network in an embodiment of the invention.

Figure 3 is a flow chart showing the steps that can be taken to upload data to a computer storage network in an embodiment of the invention;

Figure 4 is a flow chart showing the steps that can be taken to access data stored in a computer storage network in an embodiment of the invention; and

Figure 5 is a flow chart showing a sequence of steps that can be undertaken by a node to re-balance data in the network.

Detailed description of an embodiment of the invention

Figure 1 shows a decentralised computer storage network comprising a plurality of hypervisors HV1, HV2, HVn. Each hypervisor comprises one or more independent storage disks that participate in the network. For example the first hypervisor HV1 comprises a first hard disk drive 1, with a speed of 5600rpm, and a second hard disk drive 2, with a speed of 7200rpm. Hypervisor HV3 comprises a single independent storage disk 5, which is a solid state drive (SSD).

Each storage region in a hypervisor is controlled by an associated software controller. Thus, the first hard disk drive 1 in hypervisor HV1 is controlled by software controller 11 and the second hard disk drive 2 is controlled by software controller 12. The combination of a storage region and a software controller results in an individual storage node in the computer storage network.

The software controllers 11, 12 are installed in the hypervisors' existing hardware. They create virtual machines that can control all operations in the associated storage region. Specially, the software controllers are smart minimal embedded operating system (OS) controllers that manage the content stored on the associated storage region. The software controllers 11, 12 are responsible for handling input/output streams to the storage drives, and for ensuring that data are stored persistently and efficiently on the physical drive.

The software controller in each node is capable of communicating over one or more dedicated network interface cards (NICs) on the relevant hypervisor HV1. All NICs across the hypervisors are capable of communicating over the same logical subnet,

either through direct physical connection to a switch, VLAN enablement, or transparent wide area VPN membership.

All unique identifying data for a node is contained persistently on the storage drive. This means that any single storage drive is physically portable in the network. For example, the first hard disk drive 1 could be removed from hypervisor HV1 and connected to hypervisor HV2. The node would disappear from the network when it is removed. However, a new software controller would initialise in hypervisor HV2 and would associate itself with the hard disk drive 1. The new software controller could then advertise the node's new location over the network. This may be useful in disaster recovery so that a drive could be relocated if a hypervisor fails. It may also be useful during a hypervisor upgrade so that physical drives can be removed to a new location while the upgrade occurs.

Hypervisor HV1 includes portal software 20. The portal software 20 allows the hypervisor HV1 to operate as a portal node. The portal node is a gateway interface to the public internet through which an authenticated user can store and retrieve storage objects over the network.

Figure 2 is a flow chart showing the steps that are performed periodically by a node in the network. Each node is configured to perform the same sequence of steps. At step 100 the node prepares a compressed metadata file detailing a snapshot of the data stored in its associated storage drive. The metadata file includes a highly compressed summary of the data stored in the storage drive. In addition, the metadata file includes further information regarding the properties of the node such as its physical location, its access speed, its redundancy properties, and the date at which the data were last modified.

At step 102 the node is configured to broadcast the metadata file over the multicast channel so that every other node in the network can receive it. This allows each node to be aware of the state of the other nodes. At step 104 the node receives metadata broadcast from other nodes.

At step 106 the node is configured to store or update a local record of the received metadata. Thus, each node can maintain a local copy of the metadata received from every other node in the network. This can allow the software controller to make decisions concerning the distribution of data in the network. The local record is volatile and metadata updates are cached only while the relevant node is running.

Each node is configured so that it adjusts the frequency at which metadata are broadcast in step 102 in dependence on the information in the local record of received metadata. Specifically, the broadcast frequency is inversely proportional to the number of nodes from which metadata are received. In this way, the system can be arranged so that the total bandwidth required by broadcast metadata remains the same, independent of the number of nodes in the network.

Figure 3 is a flow chart showing the steps that can be taken for a user to upload data to a computer storage network. At step 200 a user can access an internet portal that will connect the user to one of the nodes in the network. The internet portal is configured to connect the user to a suitable node based on, for example, physical proximity between the user and the available nodes. Once connected to the user the selected node can behave as a 'portal' node.

A subset of nodes may be capable of behaving as 'portal' nodes. These nodes typically include a connection to a financial clearinghouse so that the user can be charged for uploading and accessing data.

At step 202 the user is connected to a portal node and the user can set their data storage requirements. For example, a user can specify that they require a certain storage capacity. At step 204 the user can specify preferences for the data to be uploaded. For example, a user may specify that they would prefer their data to be stored with high redundancy or so that it can be accessed rapidly.

At step 206 the portal node is configured to retrieve its local record of the metadata received over the network. The local record can be used to create a live picture of the state of storage nodes in the network.

At step 208 the portal node is configured to select a number of nodes, based on the local record of received metadata. The nodes may be selected by analysing the local record of received metadata in combination with the user's preferences. For example, if a user has specified that they need to access data rapidly, the portal node may select nodes that are geographically near the user and have fast data access properties. In another example, the user has specified that they require a cheaper service so the portal node may select nodes with a slower data access speed, since these nodes may charge less for data storage capacity.

The data storage devices may be categorised according to their performance requirements. For example, the access speed may be categorised as high (greater than 160MB/s), medium (greater than 100 MB/s, but less than 160MB/s), and low (less than 100MB/s).

At step 210 the portal node sends a request notification to all of the selected nodes to establish whether they are able to receive the user's data. Each node reviews the request notification and makes an independent decision about whether it is able to receive the data, responding appropriately to the portal node. The nodes that are selected for a new data set are all 'owning' nodes for that data.

At step 212 the portal node stripes and replicates the data to be uploaded. The data are striped so that they can be accessed quickly from a number of distributed nodes. The data are replicated to ensure that the data can still be accessed in its entirety should there be an error in the network. This ensures that the data storage network has an appropriate level of redundancy.

At step 214 the striped data are sent to the selected nodes so that they can be stored. An atomic transaction protocol is utilised to ensure that all end points are synchronised and the data can be stored successfully. At step 216 each node receives the striped data and analyses whether the data are already present in the storage drive. This internal de-duplication ensures that data are stored efficiently in each node.

The portal node is only used as a point of entry for the user to access the network. The identity of the portal node is not important, and any other node could fulfil this function.

Thus, there is no single central controller in the network and correspondingly there is no single point of potential failure.

Figure 4 is a flow chart showing the steps that can be taken to access data stored in the computer storage network. At step 300 a user can access an internet portal that will connect the user to one of the nodes in the network. Once connected to the user the selected node can behave as a 'portal' node.

At step 302 the portal node is configured to retrieve its local record of the metadata received over the network so that it can create a picture of all of the data files. The user can then be presented with a list of data files associated with that user. At step 304 the user can select the data files to which access is required.

At step 306 the portal node is configured to establish a connection with all synchronised nodes where relevant data is stored, using the local record of received metadata. At step 308 the user can access and/or modify data that are distributed across the storage nodes.

The portal node does not need to establish a connection with all 'owning' nodes for a particular data set at step 306. In fact, only half of the 'owning' nodes need to be available to create a full data set because of redundancy in the network. This means that data can still be recovered even if a network has been partitioned and only a subset of nodes can still communicate with one another.

Figure 5 is a flow chart showing a sequence of steps that can be undertaken by a node to re-balance data in the network. At step 400 the node retrieves its local record of received metadata. At step 402 the node determines whether a connection to one of the storage nodes has been lost. This can be achieved by analysing the time stamp of metadata received from the different nodes. If one of the node's metadata indicates an unacceptable amount of time has elapsed since the last meta-data were received then the node may instigate a re-balancing routine.

At step 406 the node commences a re-balancing routine by expelling the node to which connection has been lost. To cope with the loss of a node the data previously present

on the expelled node must be copied to a new node in the network. These data can be constructed using redundant data in the network so that the node can be fully replaced. The node is arranged to send an expulsion notification to all nodes in the network so that they can update their local record of received metadata.

When a new node is added to the network it is necessary to replicate the data on the expelled node, which can be achieved using redundant data in the network. In order to optimise efficiency in the network the data stored in the new node are analysed before any data is transferred to determine whether the stored data share any properties with the target data. In this way data transfer efficiency can be optimised by transferring only the difference data between the stored data and the target data.

At step 404 the node analyses whether all nodes in the network are properly synchronised. This is achieved by checking the date at which data were last modified for each of the relevant nodes. A node may be unsynchronised if data were last modified at an earlier time than the other nodes. If a node is found to be unsynchronised the node is expelled at step 406 and a replacement node is appointed.

Claims

1. A computer data storage network, comprising:

a plurality of storage nodes connected together to provide an integrated storage resource;

wherein each storage node is configured to broadcast metadata across the network concerning its stored data, and each storage node is configured to receive metadata that are broadcast from other storage nodes;

wherein each storage node is configured to maintain a local record of the metadata received over the network;

wherein each storage node is configured to broadcast metadata to the network repeatedly, with a predetermined frequency; and

wherein the frequency is selected in dependence on the number of storage nodes in the network according to the local record of the received metadata.

2. The computer data storage system of claim 1 wherein each storage node is configured to manage its own stored data as well as communications with other storage nodes in the network.

3. The computer data storage system of claim 1 or claim 2 wherein each storage node comprises a storage unit in which data can be stored, and a software controller for controlling read/write operations, communications with other storage nodes, and the broadcast metadata.

4. The computer data storage system of any of the preceding claims wherein one or more of the storage nodes includes an interface through which a user can upload, modify and/or access data.

5. The computer data storage system of claim 4 wherein a storage node is configured to select a number of storage nodes when new data are uploaded via the interface, wherein the selection is made using the local record of metadata received over the network.

6. The computer data storage system of claim 5 wherein the storage nodes are selected according to at least one of physical location, available data storage capacity, redundancy properties, and maximum bandwidth.

7. The computer data storage system of claim 5 or claim 6 wherein the storage node that performs the selection is also configured to upload the new data to the selected storage nodes.

8. The computer data storage system of claim 7 wherein the storage node is configured to stripe and replicate the new data before it is uploaded to the selected storage nodes.

9. The computer data storage system of any of the preceding claims wherein each storage node comprises a re-balancing module that is configured to re-balance data in the network when predetermined criteria are satisfied.

10. The computer data storage system of any of the preceding claims wherein the metadata broadcast by each storage node also includes information concerning the properties of the node.

11. The computer data storage system of any of the preceding claims wherein each storage node includes a de-duplication module that is configured to remove duplicate entries in its stored data.

12. A method of operating a computer data storage network that comprises a plurality of storage nodes connected together to provide an integrated storage resource, the method comprising the steps of:

providing independent control logic at each storage node;

broadcasting metadata across the network from each storage node concerning the data stored therein;

receiving the broadcast metadata at each storage node;

maintaining a record at each storage node concerning the metadata received over the network;

broadcasting the metadata to the network repeatedly, with a predetermined frequency; and

selecting the frequency in dependence on the number of storage nodes in the network according to the local record of the received metadata.

13. A computer readable storage medium having a computer program stored thereon, the computer program comprising:

a program module configured to provide control logic at a storage node so that the storage node can communicate with other storage nodes in a network;

a program module configured to broadcast metadata from a storage node, across the network, concerning the data stored therein;

a program module configured to receive the broadcast metadata from other storage nodes in the network; and

a program module configured to maintain a record concerning the metadata received over the network;

wherein the program module configured to broadcast metadata is configured to broadcast metadata to the network repeatedly, with a predetermined frequency; and

wherein the frequency is selected in dependence on the number of storage nodes in the network according to the local record of the received metadata.