

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
6 December 2007 (06.12.2007)

PCT

(10) International Publication Number
WO 2007/138166 A1

(51) International Patent Classification:
G06F 17/30 (2006.01)

(21) International Application Number:
PCT/FI2007/050304

(22) International Filing Date: 28 May 2007 (28.05.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
20065362 29 May 2006 (29.05.2006) FI

(71) Applicant (for all designated States except US): **TELIA-SONERA AB** [SE/SE]; -, S-106 63 Stockholm (SE).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **KOMONEN, Jussi** [FI/FI]; Aleksanterinkatu 22 D 56, FI-33100 Tampere (FI). **LEINONEN, Antti** [FI/FI]; Keltavuokonkuja 1 B 11, FI-36240 Kangasala (FI).

(74) Agent: **TAMPEREEN PATENTTITOIMISTO OY**; Hermiankatu 12 B, FI-33720 Tampere (FI).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, 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, 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, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

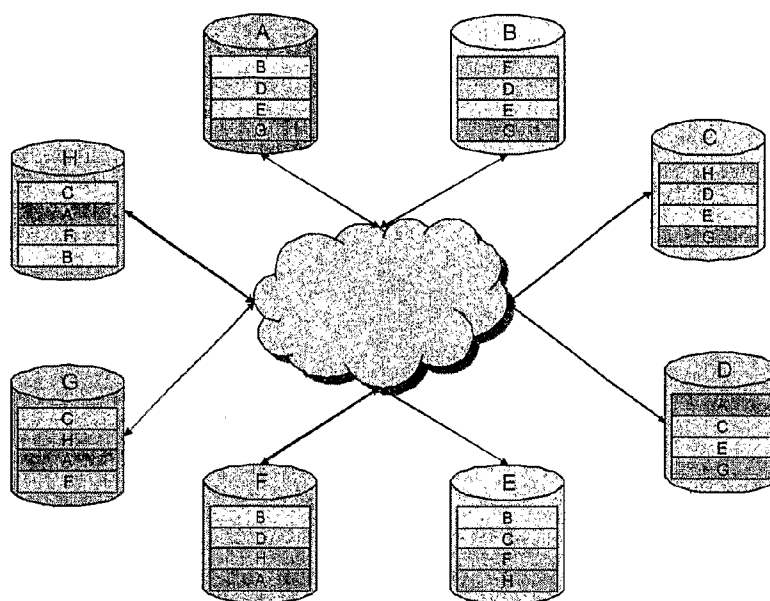
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: FILE ARCHIVES IN DISTRIBUTED FILE SYSTEM ARCHITECTURE



(57) Abstract: A distributed file system comprising a plurality of apparatuses arranged to store backup data received from other apparatuses of the distributed file system, wherein a first apparatus is arranged to fragment a data file to be archived into a plurality of subunits. The first apparatus carries out the fragmentation according to a predetermined fragmentation scheme such that the recovery of original data file requires more than one, but less than all subunits. Then the apparatus establishes peer-to-peer connections to at least two apparatuses of the distributed file system, and distributes the fragmented subunits of the data file to said at least two apparatuses to be backed up.

WO 2007/138166 A1

FILE ARCHIVES IN DISTRIBUTED FILE SYSTEM ARCHITECTURE

Field of the invention

The present invention relates to distributed file system architectures, especially when used as file archives.

5 Background of the invention

In modern world the amount of digital content is growing exponentially while the number of networked devices with own storage capacity is increasing in every household. As some examples of digital content requiring an increasing amount of storage capacity, one could mention
10 pictures taken with digital cameras or with mobile phones, video files taken with digital camcorders, or TV broadcasts stored by using PVR systems or by downloading from the Internet.

The growing amount of digital content poses the problem of archiving
15 the content. For a private person one possibility to archive the material is to store it on external physical storages, e.g. by burning the content to DVDs or storing them on external hard disks. However, such physical storages are always exposed to security risks, such as physical damage or theft. Furthermore, their suitability and durability for
20 long-term storage (for decades or even longer) has not been researched intensively yet. Another possibility to archive the material is to buy the archiving service from some network based archive service provider.

From the technical point of view, such archiving services have
25 traditionally been provided as a client-server solution, wherein the files to be backed up are transferred to some centralized server. From the service provider's point of view, the centralization of services according to classical client/server architecture allows a reasonable management
30 of complex distributed applications, assuring security, availability, and consistency. These distributed file system prototypes rely on the explicit or implicit assumption of one or more "trusted servers". A client-server approach, however, requires extensive investments into hardware and software making it expensive to deploy. Accordingly, the

increasing size of modern network infrastructures reveals the limitation of this approach to build a ubiquitous distributed file system, especially in terms of the security, the overall efficiency, the scalability, and the single point of failure problem.

5

In order to overcome the above limitations, several distributed file systems based on decentralized server architectures have been researched and developed. An article "Secure Dynamic Fragment and Replica Allocation in Large-Scale Distributed File Systems", Mei et al.,
10 IEEE Transactions on parallel and distributed systems, vol. 14, Sept 2003, (also available as <http://cesare.dsi.uniroma1.it/Sicurezza/doc/tpds.pdf>) discloses a solution based on a large number of servers coordinated by decentralized algorithms that provide the availability and scalability of the system's functionalities. There is no centralized
15 server for file system services, but a set of cooperating servers to provide data storage, whereby only clients are trusted while all servers are untrusted. Mei et al also refers to research work started at the University of Berkeley ("OceanStore", <http://oceanstore.cs.berkeley.edu/info/overview.html>) as a prototype of such distributed file system
20 architecture.

The architectures disclosed by Mei et al have, however, the significant drawback that even though the data files to be archived are fragmented and distributed over several servers, thus providing security against
25 fraudulent misuse and network and server failures, the systems still require quite extensive investments into server hardware and software in order to create commercially viable archiving service systems. Furthermore, if implemented as a commercial archiving service, the decentralized distributed file systems still have to be managed and
30 initialized centrally.

From user point of view, any kind of commercial archiving service means making contracts with a service provider about the service. Thus, there is always the question to which companies one can trust.
35 Mergers, bankruptcies and other reasons may lead to the situation where the company no longer exists which may render archive inaccessible. Accordingly, there is a need for a safe, reliable, and cost

efficient method of archiving digital content, which is especially suitable for private persons.

Summary of the invention

Now there is invented an improved distributed file system, which
5 requires no centrally managed server at all. Various aspects of the invention include a method for distributing data, a distributed file system, an apparatus and a software product, which are characterized by what is stated in the independent claim. Various embodiments of the invention are disclosed in the dependent claims.

10

According to the inventive idea, there is provided a method for distributing data in a distributed file system comprising a plurality of apparatuses arranged to store backup data received from other apparatuses of the distributed file system, the method comprising:
15 fragmenting a data file to be archived into a plurality of subunits; carrying out the fragmentation according to a predetermined fragmentation scheme such that the recovery of original data file requires more than one, but less than all subunits; establishing peer-to-peer connections from the apparatus carrying out the fragmentation to
20 at least two apparatuses of the distributed file system; and distributing the fragmented subunits of the data file to said at least two apparatuses of the distributed file system to be backed up.

25

According to an embodiment, the data file is encrypted prior to the fragmentation.

30

According to an embodiment, the original data file is restored by retrieving a necessary number of subunits from the apparatuses of the distributed file system storing the subunits as backup files, said
30 necessary number being more than one, but less than the number of all subunits.

35

According to an embodiment, in response to one of the apparatuses of the distributed file system storing the subunits being no longer
35 available, the method further comprises: selecting a new apparatus to replace said non-available apparatus as a backup partner; restoring the

original data file by retrieving the subunits of the data file from the rest of the apparatuses of the distributed file system storing the subunits; fragmenting the data file again into subunits; and distributing the fragmented subunits of the data file to a newly selected group of apparatuses of the distributed file system to be backed up.

According to an embodiment, the predetermined fragmentation scheme includes a step of parity value calculation, said parity values being usable in the recovery of original data file such that less than all subunits are required.

According to an embodiment, the predetermined fragmentation scheme is optimized according to at least one of the following attributes: the proportion of allowed storage overhead; a desired security level; available processing power; or applicable forward error correction (FEC) methods.

According to an embodiment, said apparatuses of the distributed file system comprise servers suitable for file transfer protocol (FTP); and said peer-to-peer connections between the apparatuses are established as FTP connections.

According to an embodiment, said peer-to-peer connections between the apparatuses are established as BitTorrent connections.

The arrangement according to the invention provides significant advantages. Especially from the viewpoint of private persons, using peer-to-peer connections provide the advantage that no centralized system is required for establishing the archiving system; the devices belonging to the archiving system just establish mutual connections with each other to share and retrieve the fragmented data. Thus, friends, relatives and other trusted persons can mutually decide to establish an archiving system, whereby no commercial services are required, but only an archiving software for performing the necessary actions. Carrying out the fragmentation such that more than one piece, but not necessarily all pieces, of the fragmented data are required in order to restore the original data provides the security advantage that

one cannot restore the original content by fraudulently accessing the data in one archiving storage, but on the other hand, if the data in one archiving storage is lost for some reason, the owner of the data content can restore the original data by accessing the rest of the fragmented data.

List of drawings

In the following, various embodiments of the invention will be described in more detail with reference to the appended drawings, in which

- 10 Fig. 1 illustrates an example of an archiving system according to an embodiment of the invention;
- Fig. 2 shows a simplified example of a predetermined fragmenting scheme according to an embodiment of the invention;
- 15 Fig. 3 shows another example of a predetermined fragmenting scheme according to an embodiment of the invention;
- Fig. 4 shows the fragmenting phase of the example of Fig. 3 carried out with fewer backup partners;
- 20 Fig. 5 shows a data transfer arrangement between the peers via a FTP (File Transfer Protocol) connection according to an embodiment of the invention;
- 25 Figs. 6a – 6c show another data transfer arrangements between the peers according to further embodiments of the invention;
- Fig. 7 shows a simplified block chart of a device participating in the archiving system;
- 30 Fig. 8 shows an example user interface of an archiving software implemented as a plug-in according to an embodiment of the invention.

Description of embodiments

In the following, various embodiments of the invention will be described as a mutual cooperation of a group of computer devices, whereby the users of the computer devices have agreed with each other in forehand on participating in the archiving system. The invention is, however, not limited to computers only, but it can be applied to a mobile device, set-top-box, gaming console, or media center device. Practically any IP-enabled device with local storage may act as an entity in the archive network.

The embodiments rely on two basic principles: fragmenting the data to be archived according to a predetermined fragmenting scheme and delivering the fragmented data to a plurality of archiving storages via peer-to-peer connections. The fragmentation is preferably carried out such that it is required to have more than one piece, but not necessarily all pieces, of the fragmented data in order to restore the original data. This provides the security advantage that one cannot restore the original content by fraudulently accessing the data in one archiving storage, but on the other hand, if the data in one archiving storage is lost for some reason, the owner of the data content can restore the original data by accessing the rest of the fragmented data. Using peer-to-peer connections provides the advantage that no centralized system is required for establishing the archiving system; the devices belonging to the archiving system just establish mutual connections with each other to share and retrieve the fragmented data.

The actual implementation of these features is carried out in the devices participating in the archiving system as an archiving software performing the required routines. Consequently, the software in each device keeps track of the other devices in the archiving system, carries out the fragmentation, retrieves the fragmented data from the other devices, when the original data needs to be restored, etc. The archiving software can be a separate software dedicated to a particular type of device, or if the device is a computer, the software may be a plug-in for a known software product, e.g. for Windows® Explorer.

Fig. 1 illustrates this example, wherein the archiving system consists of eight backup partners A - H. Each of the partners store fragmented data from four other partners. For example, the data of A is archived by D, F, G and H. The fragmenting scheme is defined such that if for example F is lost for some reason, the original data can still be recovered from D, G and H.

Now according to an embodiment, when the archiving software of A notices that F is no longer available as an archiving storage or it is not responding for a longer time, it inquires the other partners in the archiving system whether they have free memory available, which is allocated as an archiving storage. If for example B has enough free space, it can be selected as a new archiving partner. The original data is recovered from the fragmented data of D, G and H, the original data is fragmented again and the fragmented data previously stored by F is delivered to B. After that the data of A would be archived by D, G, H and B. Again, the original data would once more be recoverable even if one of the partners were lost.

There are several ways of splitting up the data to be backed up. For the sake of illustrating the aspects of the invention, a simple example of a predetermined fragmenting scheme is discussed herein more in detail. Fig. 2 illustrates this example, wherein the archiving system consists of nine backup partners, i.e. each participant has eight partners 1 – 8 to which the fragmented data can be shared. In this example, the original file is split into four parts by allocating every fourth character, i.e. a byte, into each part according to Fig. 2. Then each part is backed up to two partners, i.e. the first part is delivered to partners 1 and 5, the second part to partners 2 and 6, etc. Increasing the number of partners having overlapping data content will also increase the security. None of parts possessed by partners 1 - 8 alone makes sense without the other parts. Also if the original data is something else than plain text, the security effect is even greater. Thus, according to an embodiment, the original file is first encrypted and the encrypted file is backed up in the manner explained above, whereby the system becomes even more secure, virtually unbreakable.

When the data is to be restored, in this case one needs to have preferably at least four pieces of information either from 1 or 5, from 2 or 6, from 3 or 7 and from 4 or 8. The total number of suitable partner combinations for restoring is $2^4 = 16$. It should be noted that this very simple example is only meant for illustrating the basic ideas underlying the embodiments. A skilled man readily appreciates that such method, having total backup overhead of 100%, is inefficient. On the other hand, each single backup location needs to allocate space only for 25% of the original data, making it more acceptable when considering the enhanced reliability in recovering the original data.

Fig. 3 illustrates a more sophisticated parity based fragmenting scheme. Also in this example the archiving system consists of nine backup partners A – I, whereby the data of A is to be fragmented and shared among all partners. The example of Fig. 3 illustrates how an eight-character string (“A_QUICK_”) can be backed up. The original 8-bit representation of each character is put into a binary matrix, wherein backup partners B – I are allocated one bit of each original byte. Furthermore, an additional parity bit is calculated out of each byte of the original data, and this parity bit is allocated in the binary matrix to A. Out of the bits of each backup partner, a new byte to be backed up is calculated. In the example of Fig. 3, B backs up a byte with value 0, C with value 190, etc. For restoring purposes, A itself preferably backs up the byte formed of the parity bits, having the value of 109. When the parity byte is present, the original data can be restored even if one of the backup partners is lost.

The restoring phase of the example of Fig. 3 illustrates how to reconstruct the data when one backup partner, partner E in this case, is lost. First a new parity is calculated from the data from remaining backup partners B – D and F – I, in which calculation the partner containing the parity information (A) is excluded. Then an XOR operation is carried out with the newly calculated parity bits and original parity bits. The results represent the original bits of the lost backup partner E.

It should be noted that the original parity byte may also be stored by any other partner than the one whose data is to be backed up (A in this case). Then if the parity partner is the only backup partner, which is lost, there is no need to do anything special when restoring the data, i.e. if all parts of the fragmented data can be found, the parity data is not needed in restoring the original data.

The number of backup partners can also vary in parity based fragmenting schemes, whereby the original data has to be formatted accordingly. Fig. 4 shows the fragmenting phase of the previous example with five backup partners A - E. The original 8-bit representation of each character is again put into a binary matrix, wherein the 8-bit representation of each character is divided in two rows and backup partners B - E are allocated two bits of each original byte. In this case, an additional parity bit is calculated out of each row of the original data, and this parity bit is allocated in the binary matrix to A. Out of the bits of each backup partner, a new byte to be backed up is calculated for each four characters, i.e. 8 rows. In the example of Fig. 4, B backs up a byte with value 0, C with value 139, etc. for the first four characters. Then for the next four characters B backs up a byte with value 68, C with value 168, etc. A itself preferably backs up the bytes formed of the parity bits, i.e. the value of 228 for the first four characters and the value of 174 for the next four characters.

In addition to the number of backup partners, also the byte size of the original data can vary. Thereby, constructing the original data out of the data of one single backup partner is even harder.

It should be noted that the embodiments presented in Figs. 2 - 4 are just simplified examples of possible fragmenting schemes. A skilled man appreciates that a suitable fragmenting scheme is preferably evaluated case by case and more complicated fragmenting schemes may preferably be used. An optimal fragmenting scheme depends on many attributes, two most important ones being the proportion of allowed storage overhead and the desired security level. Furthermore, the available processing power or applicable forward error correction (FEC) methods, for example, have to be considered.

The archiving software, besides carrying out the fragmentation and the restoration of the original data, also takes care of the data transfer to and from the other partners of the archiving system. Like the file
5 splitting presented above, also the actual data transfer can be done in various ways.

According to an embodiment, the data transfer between the peers is carried out via a FTP (File Transfer Protocol) connection. From the
10 communication point of view, a system wherein every backup partner has an FTP server, would probably be the simplest case. Such system, comprising the backup partners A, B and C, is disclosed in Fig. 5. When the backup client needs to backup data or recover files from the backup, it initiates an FTP session and carries out the desired actions,
15 whereby the actual data transfer is performed in very simplified manner. This, however, sets some requirements for the client, i.e. the archiving software, for example in terms of password management. To be able to work in true p2p-like mode every client should know the correct username and password of every FTP server.

20 Instead of using a FTP protocol, a suitable peer-to-peer (p2p) protocol may be used for the actual data transfer. It is generally known that there exist various kinds of all-purpose p2p protocol, and additionally many dedicated p2p protocols have been developed for some specific
25 purposes. Accordingly, one can use either an existing p2p protocol or a totally new dedicated p2p protocol can be developed for the p2p backup data transfer according to the embodiments.

According to an embodiment, the data transfer can preferably be
30 carried out using a well-recognized peer-to-peer protocol called BitTorrent. BitTorrent protocol forms a good base for the p2p backup data transfer protocol. For a more complete disclosure of BitTorrent, a reference is made to BitTorrent website: <http://www.bittorrent.com/>.

35 The basic BitTorrent protocol is sufficient for transferring the files. However, traditional p2p mechanisms operate in pull mode, wherein a file is submitted in response to someone requesting it. Logically the

data backup process described above operates in an opposite way, i.e. in a push mode, wherein data is submitted to other backup partners without any request. In order to make the BitTorrent protocol applicable for the data backup process, some minor modifications are required in the normal p2p mechanism. Figures 6a – 6c show three alternative methods for implementing that functionality. For the sake of illustrating the embodiments, figs. 6a – 6c disclose A having only two backup partners, B and C, but naturally the number of the partners may vary and be significantly larger than two.

10

The first embodiment disclosed in Fig 6a is based on signalling of the new content. "A" has a file to be backed up, whereby the archiving software of A fragments the file according to a predetermined fragmenting scheme. Then A signals to B and C via some signalling mechanism that there is new data to be backed up. The signalling can be done in several ways, e.g. via a dedicated server listening for signals. When B and C have received the signalling, they will fetch the parts with normal p2p pull mechanisms.

15

20

The second embodiment disclosed in Fig 6b is based on p2p push. After the data has been split up, A pushes the new parts to B and C. For this purpose, B and C preferably comprise a server process listening for data to be pushed towards the devices B and C.

25

The third embodiment disclosed in Fig 6c is based on polling. B and C poll regularly A to notify whether there is new data to be backed up. When there is new data available at A, then B and C request it via normal p2p pull mechanisms.

30

The reconstructing of original files from the fragmented backup data can be implemented as a normal p2p pull. Thus the client A requests the backup data from the respective backup partners B and C, which submit the data in response to the request. If the data is split with explicit redundancy, like in the example of fig. 2, the data can be requested from the most suitable source.

35

The devices participating in the archiving system may be PC-based computers, known as such, connected to any data communication network, or they may be wireless terminals, like mobile stations or PDA devices, connected to any data communication network via a mobile communication network. In addition to these computing devices, the device participating in the archiving system may be any kind of electronic apparatus comprising the necessary memory to be used as the archiving storage, processing means for executing the archiving software, and communication means for transferring data between other devices participating in the archiving system.

Accordingly, a device participating in the archiving system comprises, as illustrated in Fig. 7, memory MEM, a user interface UI, I/O means I/O for arranging data transmission with other devices, and one or more central processing units CPU comprising at least one processor. The memory MEM includes a non-volatile portion for storing the applications, like the archiving software, controlling the central processing unit CPU and other data to be stored and a volatile portion to be used for temporary data processing and for operating as the archiving storage.

The steps according to the embodiments can be largely implemented with program commands executed in the central processing units CPU of the device illustrated in Fig. 7. Thus, said means for carrying out the method described above are preferably implemented as computer software code. The computer software may be stored into any memory means, such as the hard disk of a PC or a CD-ROM disc, from where it can be loaded into the memory of device. The computer software can also be loaded through a network, for instance using a TCP/IP protocol stack. It is also possible to use hardware solutions or a combination of hardware and software solutions for implementing the inventive means.

As mentioned above, the archiving software can be implemented as a plug-in for a known software product. Fig. 8 illustrates an example of a user interface of such plug-in designed for Windows® Explorer. The plug-in extends Windows® Explorer so that a backup tab is introduced. In this tab the user can select a file to be backed up, add and remove

13

backup partners, and carry out the restoration of the backed up data. Once the operation is started, the system automatically backs up the files in a background process.

- 5 It is obvious that the present invention is not limited solely to the above-presented embodiments, but it can be modified within the scope of the appended claims.

Claims:

1. A method for distributing data in a distributed file system comprising a plurality of apparatuses arranged to store backup data received from other apparatuses of the distributed file system, the method comprising:
- 5 fragmenting a data file to be archived into a plurality of subunits; **characterized** by
- 10 carrying out the fragmentation according to a predetermined fragmentation scheme such that the recovery of original data file requires more than one, but less than all subunits;
- establishing peer-to-peer connections from the apparatus carrying out the fragmentation to at least two apparatuses of the distributed file system; and
- 15 distributing the fragmented subunits of the data file to said at least two apparatuses of the distributed file system to be backed up.
2. The method according to claim 1, **characterized** by
- 20 encrypting the data file prior to the fragmentation.
3. The method according to claim 1 or 2, **characterized** by
- restoring the original data file by retrieving a necessary number of subunits from the apparatuses of the distributed file system storing the subunits as backup files, said necessary number being
- 25 more than one, but less than the number of all subunits.
4. The method according to claim 3, **characterized** by
- in response to one of the apparatuses of the distributed file system storing the subunits being no longer available, selecting a new
- 30 apparatus to replace said non-availabe apparatus as a backup partner;
- restoring the original data file by retrieving the subunits of the data file from the rest of the apparatuses of the distributed file system storing the subunits;
- fragmenting the data file again into subunits; and

distributing the fragmented subunits of the data file to a newly selected group of apparatuses of the distributed file system to be backed up.

5 5. The method according to any of the preceding claims, **characterized** by

the predetermined fragmentation scheme including a step of parity value calculation, said parity values being usable in the recovery of original data file such that less than all subunits are required.

10

6. The method according to any of the preceding claims, **characterized** by

the predetermined fragmentation scheme being optimized according to at least one of the following attributes:

15

- the proportion of allowed storage overhead;
- a desired security level;
- available processing power; or
- applicable forward error correction (FEC) methods.

20

7. The method according to any of the preceding claims, **characterized** in that

said apparatuses of the distributed file system comprise servers suitable for file transfer protocol (FTP); and

25

said peer-to-peer connections between the apparatuses are established as FTP connections.

8. The method according to any of the claims 1 - 6, **characterized** in that

30

said peer-to-peer connections between the apparatuses are established as BitTorrent connections.

9. A distributed file system comprising:

a plurality of apparatuses arranged to store backup data received from other apparatuses of the distributed file system, wherein

35

at least a first apparatus is arranged to fragment a data file to be archived into a plurality of subunits; **characterized** in that

16

said first apparatus is arranged to carry out the fragmentation according to a predetermined fragmentation scheme such that the recovery of original data file requires more than one, but less than all subunits;

5 said first apparatus is arranged to establish peer-to-peer connections to at least two apparatuses of the distributed file system; and

 said first apparatus is arranged to distribute the fragmented subunits of the data file to said at least two apparatuses of the distributed file system to be backed up.

10

10. An apparatus suitable for distributing data in a distributed file system comprising a plurality of apparatuses arranged to store backup data received from other apparatuses of the distributed file system, wherein

15

 the apparatus is arranged to fragment a data file to be archived into a plurality of subunits; **characterized** in that

 the apparatus is arranged to carry out the fragmentation according to a predetermined fragmentation scheme such that the recovery of original data file requires more than one, but less than all subunits;

20

 the apparatus is arranged to establish peer-to-peer connections to at least two apparatuses of the distributed file system; and

25 the apparatus is arranged to distribute the fragmented subunits of the data file to said at least two apparatuses of the distributed file system to be backed up.

11. The apparatus according to claim 10, **characterized** in that

30

 the apparatus is arranged to encrypt the data file prior to the fragmentation.

12. The apparatus according to claim 10 or 11, **characterized** in that

35

 the apparatus is arranged to restore the original data file by retrieving a necessary number of subunits from the apparatuses of the

distributed file system storing the subunits as backup files, said necessary number being more than one, but less than the number of all subunits.

5 13. The apparatus according to claim 12, **characterized** in that

 in response to the apparatus noticing that one of the apparatuses of the distributed file system storing the subunits is no longer available, the apparatus is arranged to select a new apparatus
10 to replace said non-available apparatus as a backup partner; whereby the apparatus is further arranged to

 restore the original data file by retrieving the subunits of the data file from the rest of the apparatuses of the distributed file system storing the subunits;

15 fragment the data file again into subunits; and
 distribute the fragmented subunits of the data file to a newly selected group of apparatuses of the distributed file system to be backed up.

20 14. The apparatus according to any of the claims 10 - 13, **characterized** in that

 the predetermined fragmentation scheme includes a step of parity value calculation, said parity values being usable in the recovery of original data file such that less than all subunits are required.

25 15. The apparatus according to any of the claims 10 - 14, **characterized** in that

 said apparatuses of the distributed file system comprise servers suitable for file transfer protocol (FTP); and

30 said peer-to-peer connections between the apparatuses are established as FTP connections.

 16. The apparatus according to any of the claims 10 - 14, **characterized** in that

35 said peer-to-peer connections between the apparatuses are established as BitTorrent connections.

17. A computer program product, stored on a computer readable medium and executable in a data processing device suitable for participating in a distributed file system, for distributing data in the distributed file system, the computer program product comprising:

5 a computer program code section for fragmenting a data file to be archived into a plurality of subunits; **characterized** in that the computer program product further comprises:

 a computer program code section for carrying out the fragmentation according to a predetermined fragmentation scheme
10 such that the recovery of original data file requires more than one, but less than all subunits;

 a computer program code section for establishing peer-to-peer connections from the data processing device carrying out the fragmentation to at least two apparatuses of the distributed file system;
15 and

 a computer program code section for distributing the fragmented subunits of the data file to said at least two apparatuses of the distributed file system to be backed up.

20 18. The computer program product according to claim 17, **characterized** in that the computer program product further comprises:

 a computer program code section for restoring the original data file by retrieving a necessary number of subunits from the apparatuses of the distributed file system storing the subunits as
25 backup files, said necessary number being more than one, but less than the number of all subunits.

 19. The computer program product according to claim 17 or 18, **characterized** in that the computer program product further
30 comprises:

 a computer program code section for monitoring the other apparatuses of the distributed file system.

 20. The computer program product according to any of the
35 claims 17 - 19, **characterized** in that the computer program product is implemented as a plug-in for another computer program product.

1/4

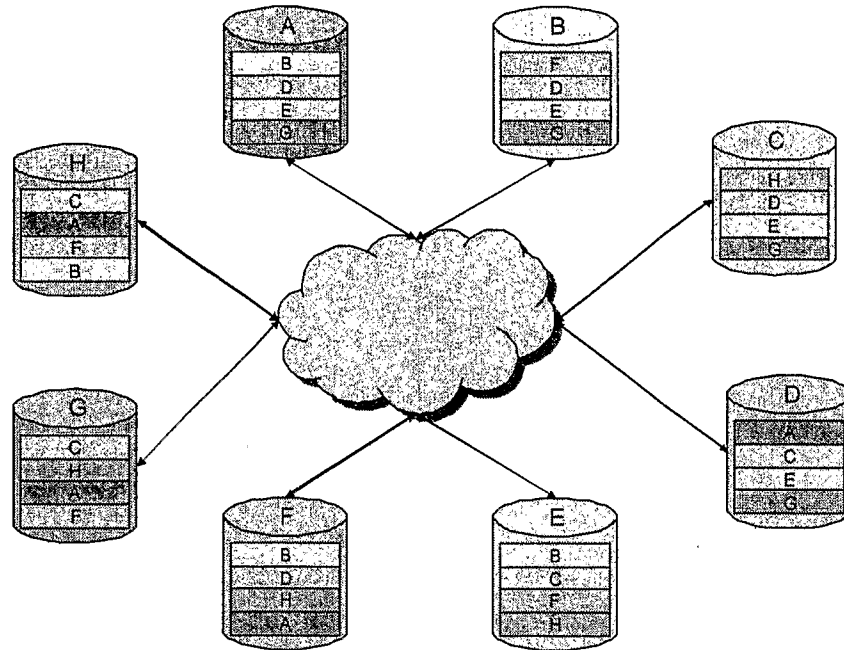


Fig. 1

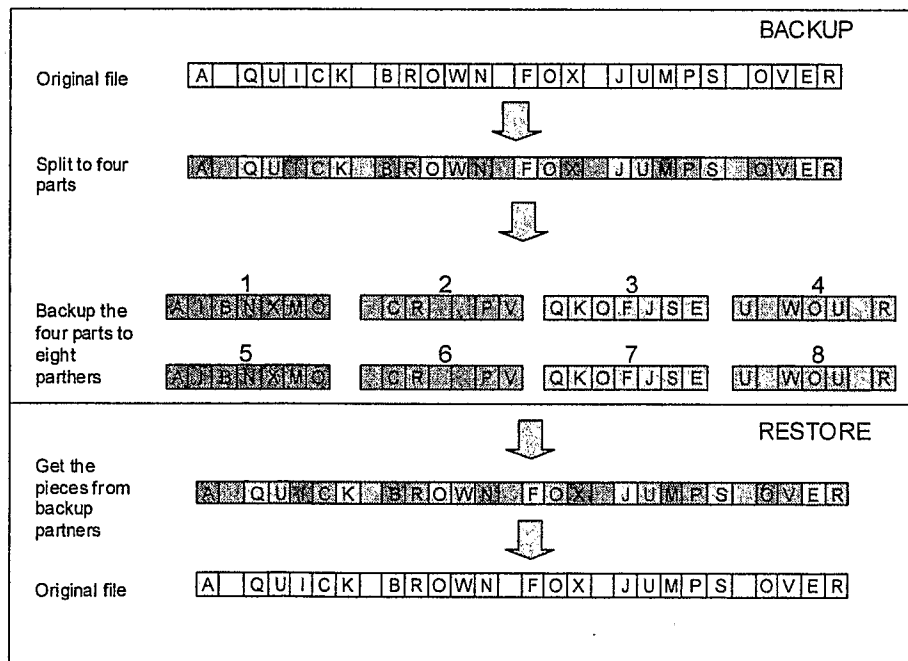


Fig. 2

2 / 4

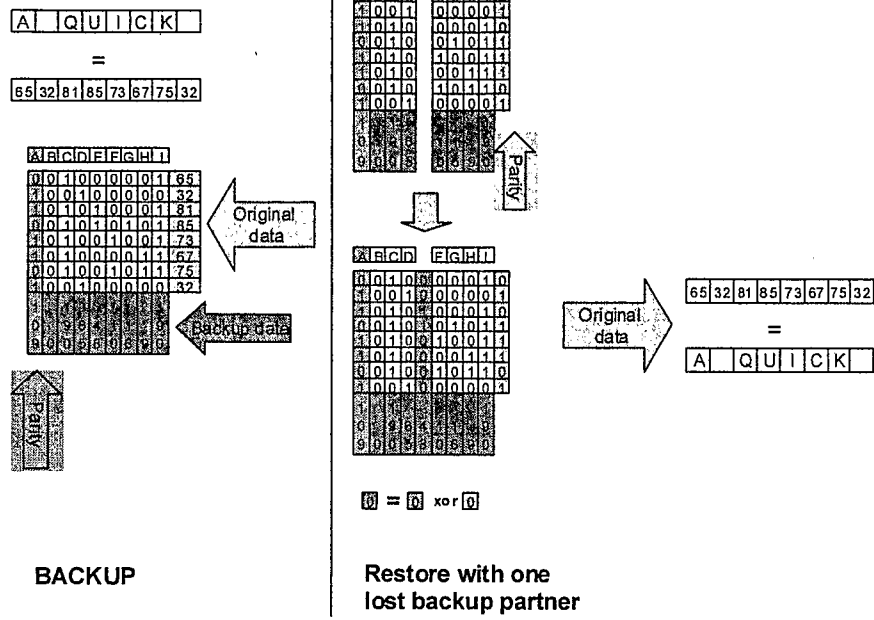


Fig. 3

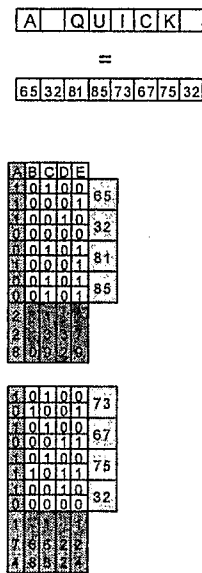


Fig. 4

3/4

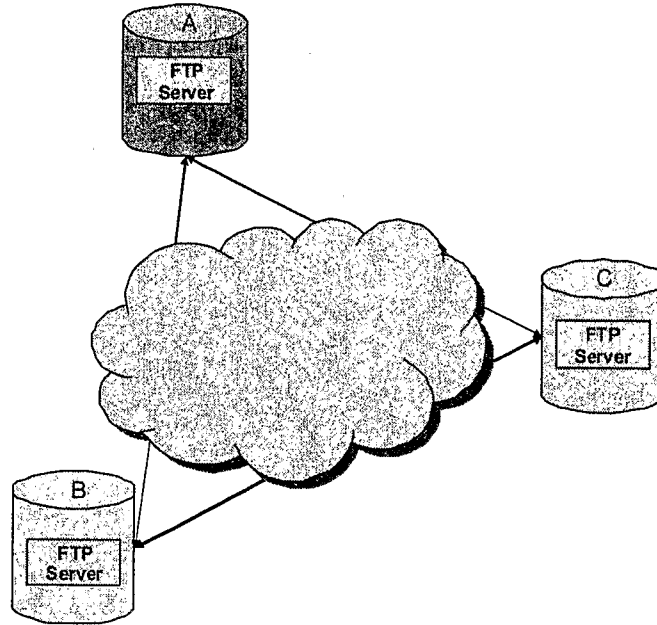


Fig. 5

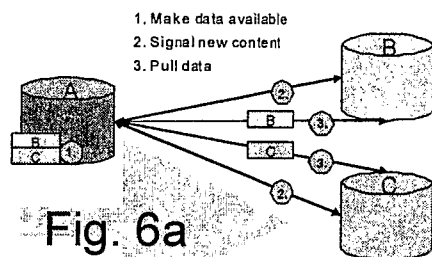


Fig. 6a

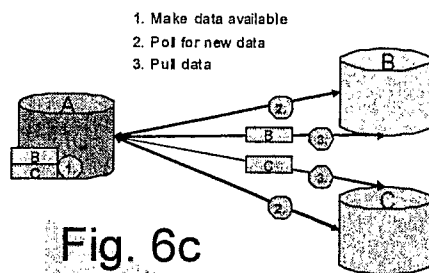


Fig. 6c

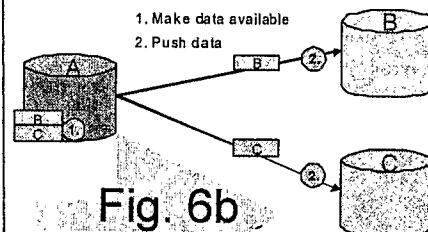


Fig. 6b

4 / 4

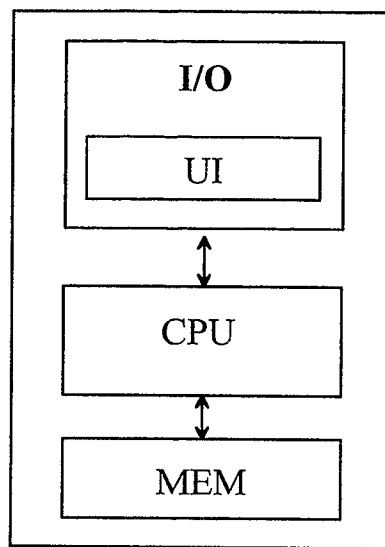


Fig. 7

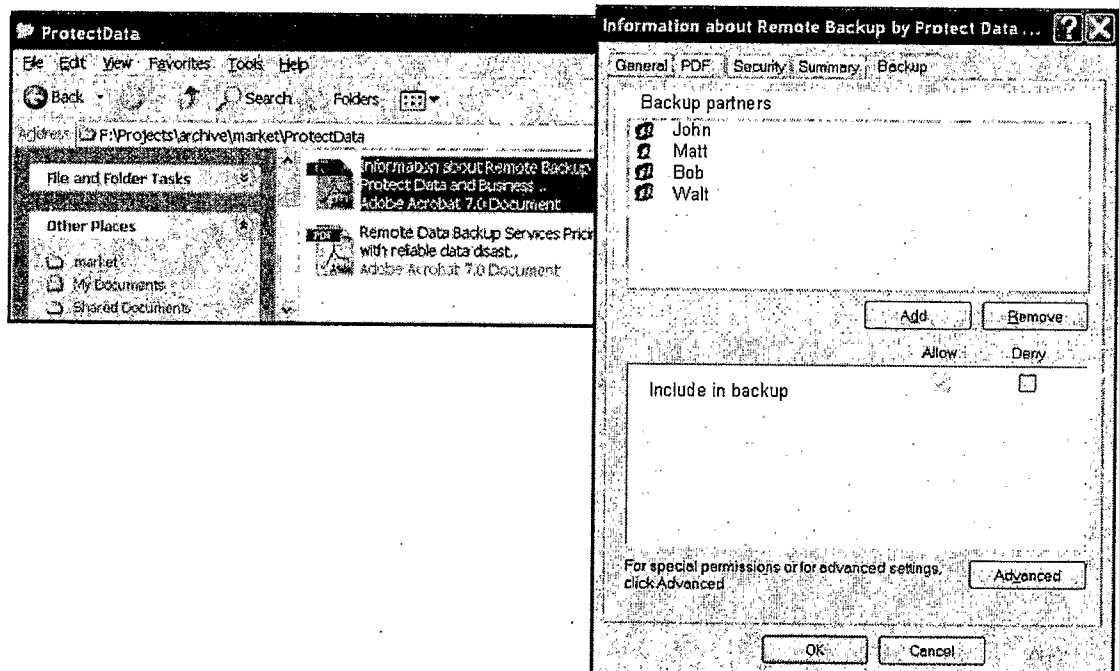


Fig. 8

INTERNATIONAL SEARCH REPORT

International application No

PCT/FI2007/050304

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	BATTEN C ET AL: "pStore: A Secure Peer-to-Peer Backup System" INTERNET CITATION, [Online] 8 December 2001 (2001-12-08), pages 1-12, XP002233336 Retrieved from the Internet: URL: http://www.cag.lcs.mit.edu/{kbarr/pstore/pStore.pdf [retrieved on 2003-03-04] abstract page 1, left-hand column, line 22 - page 1, right-hand column, line 25 page 3, left-hand column, line 9 - page 5, right-hand column, line 41 page 6, left-hand column, line 5 - page 6, right-hand column, line 17 ----- -/--	1-20

☒ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

G document member of the same patent family

Date of the actual completion of the international search

17 August 2007

Date of mailing of the international search report

30/08/2007

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Boyadzhiev, Yavor

INTERNATIONAL SEARCH REPORT

International application No

PCT/FI2007/050304

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	MEI A ET ALL: "Secure Dynamic Fragment and replica allocation in large-scale distributed file systems" IEEE, TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, [Online] vol. 14, no. 9, September 2003 (2003-09), pages 885-896, XP002447123 Published by the IEEE Society Retrieved from the Internet: URL: http://ieeexplore.ieee.org/iel5/71/27643/01233711.pdf [retrieved on 2007-08-16] cited in the application abstract page 885, right-hand column, line 7 - page 886, right-hand column, line 54 page 887, left-hand column, line 4 - page 887, left-hand column, line 8 page 888, left-hand column, line 12 - page 889, left-hand column, line 63 page 891, right-hand column, line 5 - page 891, right-hand column, line 29 page 892, left-hand column, line 28 - page 892, left-hand column, line 59	1-20
X	US 2004/153458 A1 (NOBLE BRIAN D [US] ET AL) 5 August 2004 (2004-08-05) abstract paragraph [0003] - paragraph [0009] paragraph [0019] - paragraph [0028] paragraph [0031] - paragraph [0039] paragraph [0048] - paragraph [0050] paragraph [0053] - paragraph [0055] paragraph [0062] - paragraph [0076] paragraph [0079] paragraph [0084] paragraph [0093] - paragraph [0095] paragraph [0101] - paragraph [0102] paragraph [0110]	1-20
A	CHAN J S K ET AL: "Scheduling Algorithms for Peer-to-Peer Collaborative File Distribution" COLLABORATIVE COMPUTING: NETWORKING, APPLICATIONS AND WORKSHARING, 2005 INTERNATIONAL CONFERENCE ON SAN JOSE, CA, USA 19-21 DEC. 2005, PISCATAWAY, NJ, USA, IEEE, 19 December 2005 (2005-12-19), pages 1-10, XP010926534 ISBN: 1-4244-0030-9 abstract page 1, left-hand column, line 24 - page 1, right-hand column, line 41 page 3, right-hand column, line 34 - page 4, left-hand column, line 27	1-20

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/FI2007/050304

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2004153458	A1	05-08-2004	NONE