

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
21 December 2000 (21.12.2000)

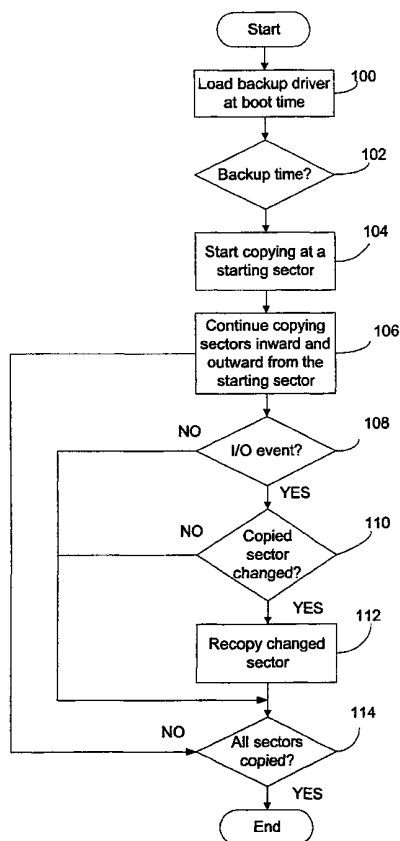
PCT

(10) International Publication Number
WO 00/77641 A1

- (51) International Patent Classification⁷: G06F 11/14
- (21) International Application Number: PCT/US00/16413
- (22) International Filing Date: 14 June 2000 (14.06.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data: 09/333,741 15 June 1999 (15.06.1999) US
- (71) Applicant: MICROSOFT CORPORATION [US/US];
One Microsoft Way, Redmond, WA 98052 (US).
- (72) Inventors: WATSON, Brandon, L.; 16918 SE 38th Place,
Bellevue, WA 98008 (US). GASCH, Scott, M.; 12626
NE 114th Place, Kirkland, WA (US). GUITTET, Michel;
11014 167th Court, Redmond, WA 98052 (US).
- (74) Agents: CHANG, Y., Kurt et al.; Leydig, Voit & Mayer,
Ltd., Two Prudential Plaza, Suite 4900, 180 North Stetson,
Chicago, IL 60601-6780 (US).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU,
AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE,
DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU,
ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS,
LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO,
NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR,
TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM,
KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian
patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European
patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,
IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG,
CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:
— With international search report.

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR GENERATING A BACKUP COPY OF A STORAGE MEDIUM



(57) Abstract: A backup driver generates a backup copy of a main storage medium, such as a hard disk, on a backup storage medium without interfering with access to the files on the main storage medium by applications running on the computer system. The backup driver is loaded at boot time to gain access to the main storage medium on a sector level that is not blocked by the read-write exclusive locks imposed on the files on the main storage medium by the applications. When a scheduled backup time arrives, the backup driver begins to copy the main storage medium sector-by-sector to the backup storage medium. At the same time, the backup driver monitors I/O events on the main storage medium to see whether the I/O events change any sector within the range of sectors already copied to the backup storage medium. If a sector within that range of copied sectors is changed, the backup driver recopies that sector to the backup storage medium to ensure consistency between the main storage medium and the backup storage medium. The backup driver continues to copy the sectors on the main storage medium to the backup storage medium until all sectors on the main storage medium are copied, at which time the backup operation is complete. Because the backup operation is performed on a sector level rather than on a file level, it does not depend on the file structures on the main storage medium.



-
- *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*
- 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.*

SYSTEM AND METHOD FOR GENERATING A BACKUP COPY OF A STORAGE MEDIUM

5

TECHNICAL FIELD

This invention relates generally to data backup and, more particularly, relates to the generation of a backup copy of a data storage medium, such as a hard disk, in a computer system without interfering with access to the files on the storage medium by applications running on the computer system.

10

BACKGROUND OF THE INVENTION

A computer system typically has non-volatile mass storage media, such as one or more hard disks, for storing data files and software applications. In connection with the fast pace of developments of computer technologies, the capacities of hard disks used in common computer systems have increased significantly over the years. Even for personal computers, it is common for a modern hard disk to have a storage capacity on the order of tens of giga-bytes. As the storage capacities of the hard disks grow larger, the risk of accidentally losing critical data or applications on a disk due to problems such as disk failure or other catastrophic system failure becomes ever more significant. To avoid a total loss of the information stored on a hard disk, it is important to periodically copy, or backup, the contents of the hard disk on another storage medium, which may be, for example, a magnetic tape, a magneto-optical disk, or another hard disk, etc. This copy, typically referred to as a backup copy or backup image, may be used to reconstruct the data on the hard disk if the data on the hard disk should be corrupted or become inaccessible for some reason.

The purpose of the backup process is to generate a backup copy that is a faithful snapshot of files on the hard disk being copied at the time the backup process is completed. Because it is not practical in many applications to allocate a time period dedicated for the backup operation, preferably the backup copy of a hard disk can be created while the computer system is in operation. It has been difficult, however, to provide a backup process that is both reliable and efficient without affecting, or being affected by, the access to the files on the disk by applications running on the computer system. When the computer system is in operation, the files on the hard disk are constantly being accessed by the applications. Thus, while the backup process is attempting to create a snapshot of the disk, the contents of the files are constantly being changed. When the backup generates a backup copy of a file, it must read the file in and copy the bits of the file to the backup storage medium. If the file on the disk being copied is not locked from read-write access by the applications, an application may change one portion of the file while another portion of the file is being copied for backup. In that case, the backup process is actually trying to hit a moving target. By the time the backup process finishes copying the file, the backup file may not correspond to the file on the hard disk.

It is possible to block all access to the hard disk by the applications during the backup process to ensure that the files are not modified by the applications during the backup process. In most cases, however, such an approach is not acceptable because the operation of the applications cannot be interrupted by the backup process, especially when the hard disk has a large storage capacity that takes a long time to be copied. Without the ability to lock the files, however, the backup process cannot ensure that the

files being copied to the backup medium are not changed during the copying process. As a result, the reliability of the backup copy cannot be guaranteed.

A related problem encountered in a backup process is that an application may lock up one or more files for exclusive read-write access, and those locked files cannot be
5 accessed by other applications, including the backup program. During a backup operation, if the backup program encounters a file that has been locked up by another application, it may have to halt the backup process until the application releases the lock on the file. This approach, however, can delay the backup process indefinitely. Many backup programs currently available try to handle this problem by skipping files opened
10 by other applications and attempting to come back to those skipped files later when they are no longer locked. Although those backup programs have different degrees of flexibility in handling open files, they are unsatisfactory in that the backup process cannot be completed as long as there is one application that maintains an exclusive lock on a single file on the disk being copied.

15

SUMMARY OF THE INVENTION

In accordance with the invention, there is provided a system and method that uses a backup driver for copying data from a main storage medium, such as a hard disk, to a backup storage medium to generate a backup copy of the main storage medium. The
20 backup driver has access to the main storage medium on a sector level that is not blocked by read-write exclusive locks imposed on the files on the main storage medium by applications running on the computer system. During a backup operation, the driver copies the main storage medium sector by sector to the backup storage medium. At the

same time, the backup driver monitors I/O events on the main storage medium to see whether any I/O event changes any sector within the range of sectors already copied to the backup medium. If a sector within that range of copied sectors is changed, the driver recopies that changed sector to the backup storage medium. The driver continues to copy
5 the sectors on the main storage medium to the backup storage medium until all sectors on the main storage medium have been copied, at which time the backup process is complete.

Additional features and advantages of the invention will be made apparent from the following detailed description of illustrative embodiments which proceeds with
10 reference to the accompanying figures.

BRIEF DESCRIPTION OF THE DRAWINGS

While the appended claims set forth the features of the present invention with particularity, the invention, together with its objects and advantages, may be best
15 understood from the following detailed description taken in conjunction with the accompanying drawings of which:

Figure 1 is a block diagram generally illustrating an exemplary computer system on which the present invention resides;

FIGS. 2A and 2B are schematic diagrams showing alternative arrangements of a
20 main disk and a backup disk;

FIG. 3 is a schematic diagram showing sectors on a main disk being copied to a backup disk;

FIG. 4 is a schematic diagram showing an embodiment of a system for generating a reliable backup copy of a storage medium according to the invention; and

FIG. 5 is a flowchart showing steps for backing up a storage medium according to an embodiment of the invention.

5

DETAILED DESCRIPTION OF THE INVENTION

Turning to the drawings, wherein like reference numerals refer to like elements, the invention is illustrated as being implemented in a suitable computing environment. Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by a personal
10 computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multi-
15 processor systems, microprocessor based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and
20 remote memory storage devices.

With reference to Fig. 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a conventional personal computer 20, including a processing unit 21, a system memory 22, and a system bus 23

that couples various system components including the system memory to the processing unit 21. The system bus 23 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes read only memory (ROM) 24 and
5 random access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the basic routines that help to transfer information between elements within the personal computer 20, such as during start-up, is stored in ROM 24. The personal computer 20 further includes a hard disk drive 27 for reading from and writing to a hard disk 60, a magnetic disk drive 28 for reading from or writing to a removable magnetic disk 29, and
10 an optical disk drive 30 for reading from or writing to a removable optical disk 31 such as a CD ROM or other optical media.

The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive interface 33, and an optical disk drive interface 34, respectively. The drives and their
15 associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for the personal computer 20. Although the exemplary environment described herein employs a hard disk 60, a removable magnetic disk 29, and a removable optical disk 31, it will be appreciated by those skilled in the art that other types of computer readable media which can store data
20 that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories, read only memories, and the like may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk 60, magnetic disk 29, optical disk 31, ROM 24 or RAM 25, including an operating system 35, one or more applications programs 36, other program modules 37, and program data 38. A user may enter commands and information into the personal computer 20 through input devices
5 such as a keyboard 40 and a pointing device 42. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 21 through a serial port interface 46 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port or a universal serial bus (USB). A monitor 47 or other
10 type of display device is also connected to the system bus 23 via an interface, such as a video adapter 48. In addition to the monitor, personal computers typically include other peripheral output devices, not shown, such as speakers and printers.

The personal computer 20 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 49. The
15 remote computer 49 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the personal computer 20, although only a memory storage device 50 has been illustrated in Fig. 1. The logical connections depicted in Fig. 1 include a local area network (LAN) 51 and a wide area network (WAN) 52. Such
20 networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the personal computer 20 is connected to the local network 51 through a network interface or adapter 53. When used

in a WAN networking environment, the person computer 20 typically includes a modem 54 or other means for establishing communications over the WAN 52. The modem 54, which may be internal or external, is connected to the system bus 23 via the serial port interface 46. In a networked environment, program modules depicted relative to the
5 personal computer 20, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

In the description that follows, the invention will be described with reference to acts and symbolic representations of operations that are performed by one or more
10 computer, unless indicated otherwise. As such, it will be understood that such acts and operations, which are at times referred to as being computer-executed, include the manipulation by the processing unit of the computer of electrical signals representing data in a structured form. This manipulation transforms the data or maintains it at locations in the memory system of the computer, which reconfigures or otherwise alters the operation
15 of the computer in a manner well understood by those skilled in the art. The data structures where data is maintained are physical locations of the memory that have particular properties defined by the format of the data. However, while the invention is being described in the foregoing context, it is not meant to be limiting as those of skill in the art will appreciate that various of the acts and operation described hereinafter may
20 also be implemented in hardware.

The present invention is directed to a system and method that enables the generation of a reliable backup copy of the contents of a storage medium, such as a hard disk, while the computer system is in operation, without interfering with access to the

files on the storage medium by applications running on the computer system. The present invention avoids the file-locking problem experienced by conventional backup methods by copying the storage medium on a sector basis rather than on a file basis. For illustration purposes, the invention will be described below in connection with a preferred embodiment in which a main disk is copied to a backup disk. It will be appreciated, however, the present invention is not limited to copying from or to a storage disk but is applicable to all types of storage media that support read/write access of stored data on a sector level.

Referring to FIGS. 2A and 2B, in accordance with a preferred embodiment of the invention, a backup driver 70 is provided to handle a backup process in which the contents of a main disk 72 is copied to a backup disk 74 to generate a backup copy. As shown in FIG. 2A, the main disk 72 and the backup disk 74 may be two separate physical devices. Alternatively, the main disk 72 and the backup disk 74 may correspond to two partitions on the same physical device, as shown in FIG. 2B. As another alternative, the main disk may be a partition on one physical device, and the backup disk may be a partition on another physical device.

In accordance with an aspect of the invention, instead of trying to copy the main disk 72 on a file-by-file basis for backup, the backup driver 70 copies the main disk sector by sector to the backup disk 74. To that end, the backup driver 70 has access to the main disk 72 on a sector level that is not hindered by any read-write exclusive lock on the files of the main disk imposed by applications accessing the main disk. In a preferred embodiment, to give the backup driver such sector-level access, the backup driver is loaded at boot time of the computer system, i.e., at the time the computer is powered up,

and stays loaded on the computer system. This allows the backup driver to lock the main disk on a sector level before there are open file handles active for the disk. This condition is available immediately following the loading of the file system during a normal boot process. Otherwise, there are likely to be open system files, such as those of a security
5 database, that would prevent the backup driver from locking the disk on the sector level.

With the sector-level access to the main disk, the backup driver 70 can access a sector on the main disk at any time during the operation of the computer system, even if that sector is in a file under a read-write exclusive lock by an application. Because the access of the main disk by the backup driver is not constrained by the use of the files by
10 the applications, the backup process can be initiated any time during the operation of the computer system. The backup process also does not require that the main disk be locked from other applications to prevent changes made to the files on the main disk during the backup operation. As a result, the backup process has no effect on the accessibility of the files to the applications running on the computer system.

15 During the backup process, because the files on the main disk 72 are not locked from read-write access by the applications, it is possible that before all the sectors on the main disks are copied to the backup disk 74, an application will change one or more sectors on the main disk that have already been copied to the backup disk. When such changes take place, the entire image on the backup disk will become useless if it is not
20 updated. This is because the backup process copies sectors rather than files. To make the backup copy readable, the backup disk should match the main disk exactly on the sector level. If such match is not maintained, there could be file pointers that point to old

sectors with old data, causing a read failure. Thus, one bad sector could render the entire backup image useless.

To guarantee consistency between the sectors on the main disk and those on the backup disk, the backup driver continuously monitors low-level input/output (I/O) events on the main disk during the backup process. In one embodiment, the file system initiates the write operation to the main disk and passes a byte offset from the beginning of the disk and a length of the data to be written. Each of these two values is a multiple of the sector length. These two numbers allow the backup driver to determine which sectors are affected by the write operation. If an I/O event changes one of the sectors already copied onto the backup disk, the backup driver retrieves the current contents of that sector on the main disk and write them into a corresponding sector on the backup driver. In other words, the backup driver recopies those sectors that have already been copied to the backup disk but are later modified by I/O events during the backup process. In this way, the backup driver ensures that all of the sectors on the backup driver are kept up-to-date so that at the end of the backup process the backup disk contains an exact copy or image of the contents of the main disk at that moment.

By way of example, FIG. 3 shows a schematic view of the main disk 72 and the backup disk 74 on which a backup copy of the main disk is to be generated. The main disk 72 and the backup disk 74 each has multiple tracks, and each track has multiple sectors. In a preferred embodiment, the backup process starts by selecting a sector on the main disk, such as the sector 80 on the track 78, as a starting point, and copies that sector into a corresponding sector 82 on the backup disk. Exactly which sector is selected as the starting point is not critical, although preferably the selected sector is close to the center

of the volume of the main disk. Thereafter, the backup driver copies the sectors inward and outward of the starting sector, and so on, such that the boundaries, or “watermarks,” of the range of sectors copied to the backup disk extend inward and outward from the starting point. As described above, during this copying process, the backup driver

5 monitors all low-level I/O events on the main disk and updates the backup disk if a low-level I/O event modifies any sector that falls within the watermarks on the main disk. For example, if the backup driver has copied sectors on the main disk up to the watermarks 84, 86, and a low-level I/O event changes the sector 88 that is within the watermarks, the backup driver recopies the modified sector 88 into the corresponding sector 89 on the

10 backup disk 74. The backup driver continues to copy the sectors on the main disk until the ends of the volume of the main disk are reached, i.e., all sectors on the main disk have been copied onto the backup disk. At that time, the backup disk will be an exact copy of the main disk, and the backup process is complete.

Because the backup copying is on a sector basis instead of a file basis, empty

15 sectors as well as non-empty sections on the main disk would be copied to the backup disk. With a straightforward sector-to-sector copying, the time required to copy a disk does not depend on the number or sizes of files or the total amount of data stored on the main disk. Optimization, however, may be made by identifying unused sectors on the main disk and skipping such sectors during the backup process. It should also be noted

20 that the amount of time required to complete the backup process will depend on the read/write events taking place during the backup process. This is because the more read/write events occur during the backup process, the more likely the backup driver would have to recopy sectors that have already been copied.

In a preferred embodiment, the backup system according to the invention is implemented in the framework of the Microsoft Windows NT operating system. As shown in FIG. 4, the backup driver 70 is a loadable device driver that runs in the kernel mode. In the illustrated architecture, there are three layered drivers for handling I/O access to the main disk and the backup disk, including a File System (FS) driver 90, the backup driver 70, and a disk driver 92. The layered drivers pass I/O requests to one another by calling the I/O manager. Relying on the I/O manager as an intermediary allows each driver to maintain independence so that it can be loaded or unloaded without affecting other drivers. The system may also include other drivers for added data protection. For instance, FIG. 4 shows a Fault Tolerant driver 96 on the same level with the backup driver. The Fault Tolerant driver 96 is used to provide data redundancy by duplicating, or "mirroring" the main disk onto another hard disk such that the data on the main disk and the mirror disk are exactly identical at all times.

The backup process of a preferred embodiment is summarized in the flowchart of FIG. 5. At the boot time of the computer, i.e., when the computer is powered up, the backup driver is loaded (step 100). As described above, loading the backup driver at boot time allows the driver to gain access to the entire main disk on a sector level, such that the backup driver can read the sectors on the main disk at any time without being blocked by any read-write exclusive locks imposed on the files on the hard disk. In preferred embodiment, the backup process is triggered periodically upon the request of a backup application. When a scheduled time for backup arrives (step 102), the backup application sends an instruction to backup to the backup driver 70. Alternatively, the user may initiate the backup operation by giving an appropriate command. Upon receiving the

instruction to backup, the backup driver selects a sector on the main disk as the starting point for backup (step 104), and starts to copy the main disk sector by sector inward and outward from the starting sector to the backup disk (step 106).

As the copying process goes on, the backup driver monitors low-level I/O
5 operations to the main disk (step 108). If a low-level I/O event changes a sector that falls within the "watermarks," i.e., a sector that has already been copied to the backup disk (step 110), the backup driver retrieves the contents of that altered sector on the main disk and writes them to the corresponding sector on the backup disk (step 112). In this way, the data on the backup disk is constantly updated to reflect the changes made to the main
10 disk during the backup process. The backup driver continues to copy the sectors on the main disk to the backup disk until the ends of volume of the main disk are reached (step 114). At that time, the backup process is complete, and the backup disk contains a faithful copy, or a snapshot, of the contents of the main disk. The backup driver then awaits the next backup instruction, which will be given at the next scheduled backup
15 time, to repeat the backup process to generate a new backup copy on the backup disk.

As described above, the backup process does not interfere with the accessibility of the files on the main disk by other applications. Moreover, because the backup driver is loaded at boot time and is therefore available any time during the operation of the computer system, the backup operation is performed on the fly without the need for
20 rebooting the system.

Although the invention has been described above by referring to a preferred embodiment in which both the main storage medium and the backup storage medium are disks, it will be appreciated that the invention can be implemented with other types of

storage media that provide section-level data access. By way of example, a magnetic tape may be used to backup a hard disk. In that case, recopying a sector may require the rewinding of the backup tape. As another example, the backup medium may be a write-once storage medium, such as a magneto-optical disk. Sectors changed during the backup process could be appended to the existing backup image on the write-once medium. Of course, if there is a high level of I/O activity on the main storage medium during the backup process, the write-once backup medium would need a capacity sufficiently larger than that of the main storage medium in order to accommodate all recopied sectors.

In view of the many possible embodiments to which the principles of this invention may be applied, it should be recognized that the embodiment described herein with respect to the drawing figures is meant to be illustrative only and should not be taken as limiting the scope of invention. For example, those of skill in the art will recognize that the elements of the illustrated embodiment shown in software may be implemented in hardware and vice versa or that the illustrated embodiment can be modified in arrangement and detail without departing from the spirit of the invention. Therefore, the invention as described herein contemplates all such embodiments as may come within the scope of the following claims and equivalents thereof.

What is claimed is:

1. In a computer system having a main storage medium and a backup storage
5 medium each having a plurality of sectors, a method of copying contents of the main
storage medium to the backup storage medium to create a backup copy on the backup
storage medium, comprising the steps of:

obtaining sector-level access to the main storage medium;

upon receipt of a backup instruction, initiating a copying process to copy contents
10 of the sectors on the main storage medium into corresponding sectors on the backup
storage medium;

monitoring low-level input/output (I/O) events on the main storage medium to
determine whether a low-level I/O event modifies a sector on the main storage medium
that have been copied to the backup storage medium;

15 for a sector on the main storage medium that have been copied to the backup
storage medium and then modified by the low-level I/O event, recopying said modified
sector to a corresponding sector on the backup storage medium; and

terminating the copying process when all sectors on the main storage medium are
copied onto the backup storage medium.

20

2. A method as in claim 1, wherein the main storage medium is a hard disk.

3. A method as in claim 2, wherein the backup storage medium is a hard disk.

4. A method as in claim 1, wherein the copying process includes:

copying a starting sector on the main disk to the backup storage medium;

continuing to copy sectors on the main storage medium such that a range of

5 sectors on the main storage medium that have been copied to the backup storage medium
expands inward and outward from the starting sector.

5. A method as in claim 1, wherein the main storage medium and backup storage

medium are separate physical devices.

10

6. A method as in claim 1, wherein the main storage medium and backup storage

medium are partitions of one physical device.

7. A method as in claim 1, wherein the step of obtaining sector level access

15 includes loading a backup driver at boot time of the computer system.

8. A computer-readable medium having computer executable instructions for

performing steps to backup a main storage medium onto a backup storage medium,

comprising:

20 accessing the main storage medium on a sector level;

copying sectors of the main storage medium onto the backup storage medium;

monitoring, during copying, low-level input/output (I/O) events to the main storage medium to determine whether a low-level I/O event modifies a sector on the main storage medium that has been copied to the backup storage medium; and

when a low-level I/O event modifies a sector that has been copied to the backup
5 storage medium, recopying said modified sector onto the backup storage medium.

9. A computer-readable medium as in claim 8, wherein the step of copying includes:

copying a starting sector on the main storage medium to the backup storage
10 medium;

continuing to copy sectors on the main storage medium such that a range of copied sectors extends inward and outward from the starting sector.

10. A computer system comprising:

15 a main storage medium having a plurality of sectors;

a backup storage medium having a plurality of sectors;

a backup driver loaded to have sector-level access to the main storage medium,
the backup driver upon receipt of a backup instruction initiating a copying process to
copy the sectors on the main storage medium to corresponding sectors on the backup
20 storage medium, monitoring low-level input/output (I/O) access to the main storage
medium and, when the low-level I/O access modifies a sector within a range of sectors on
the main storage medium that have been copied to the backup storage medium, copying
said modified sector to a corresponding sector on the backup storage medium.

1/5

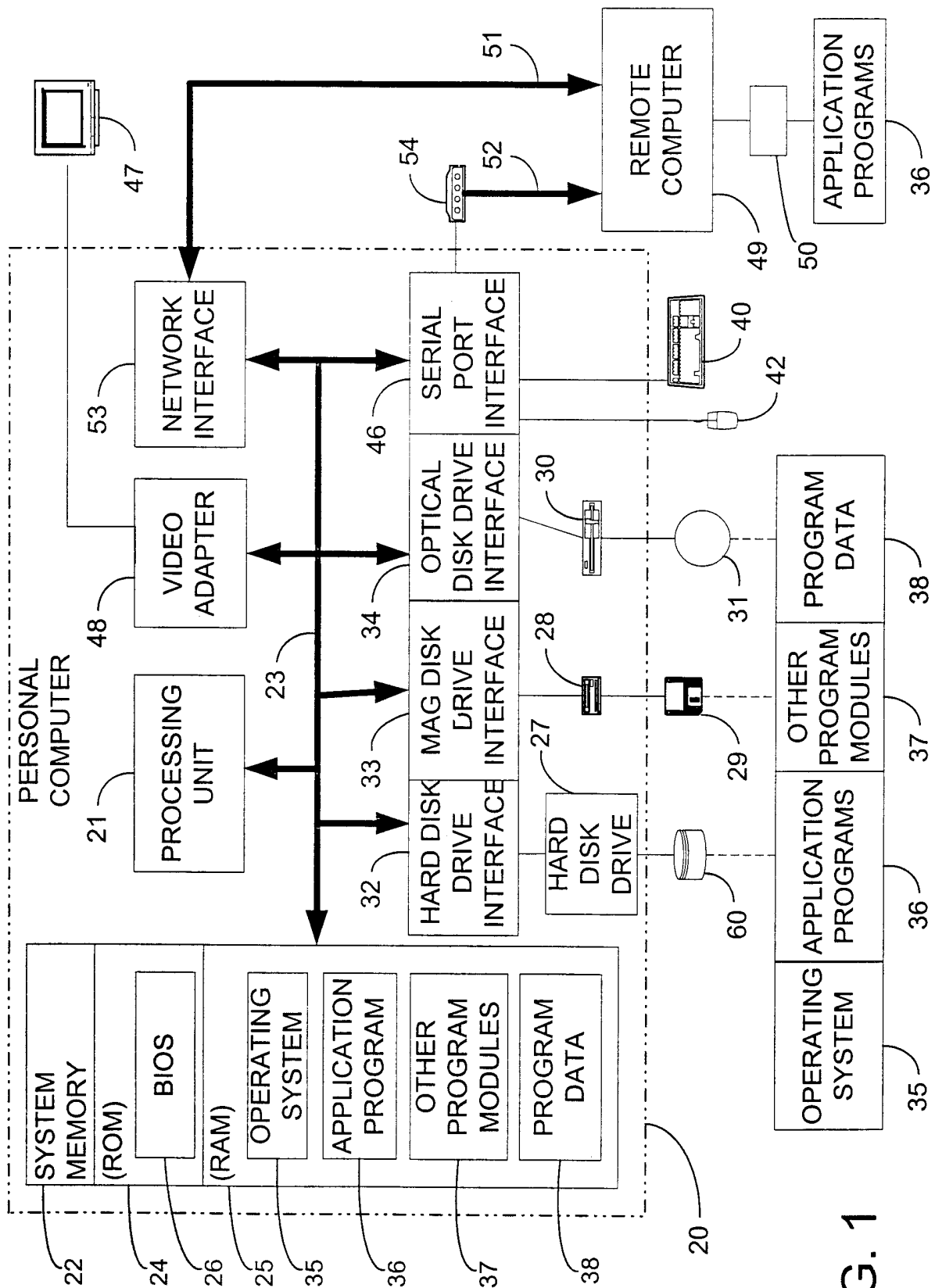


FIG. 1

2/5

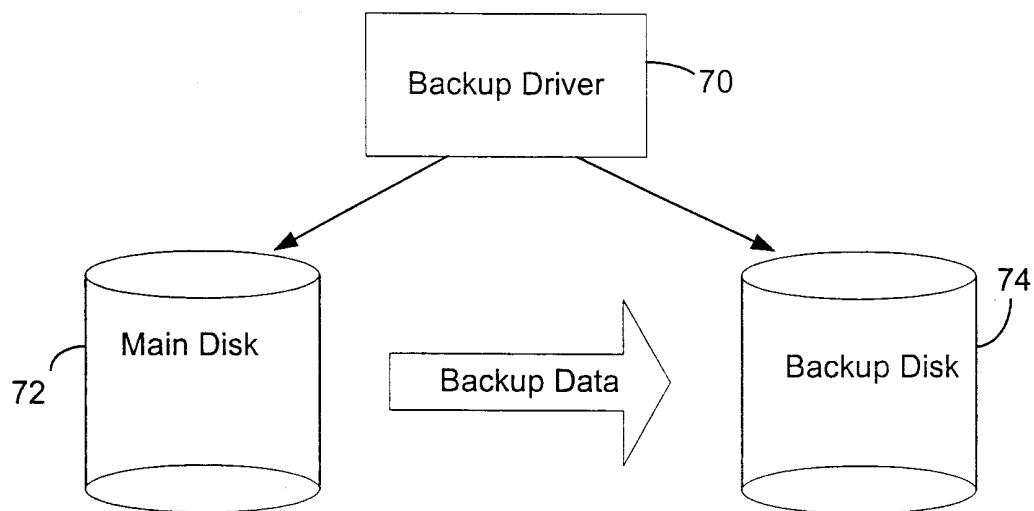


FIG. 2A

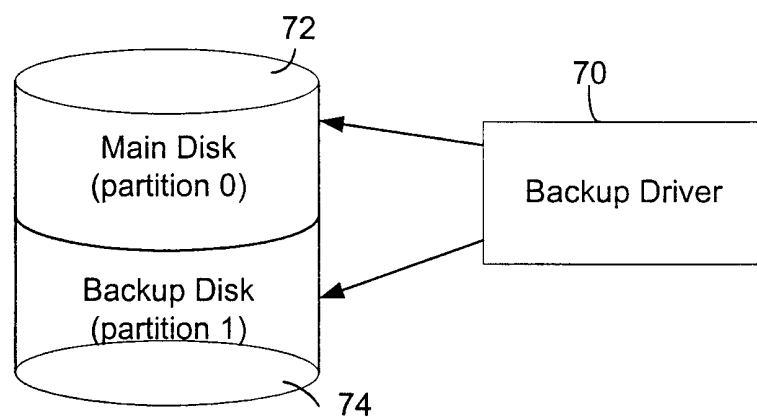


FIG. 2B

3/5

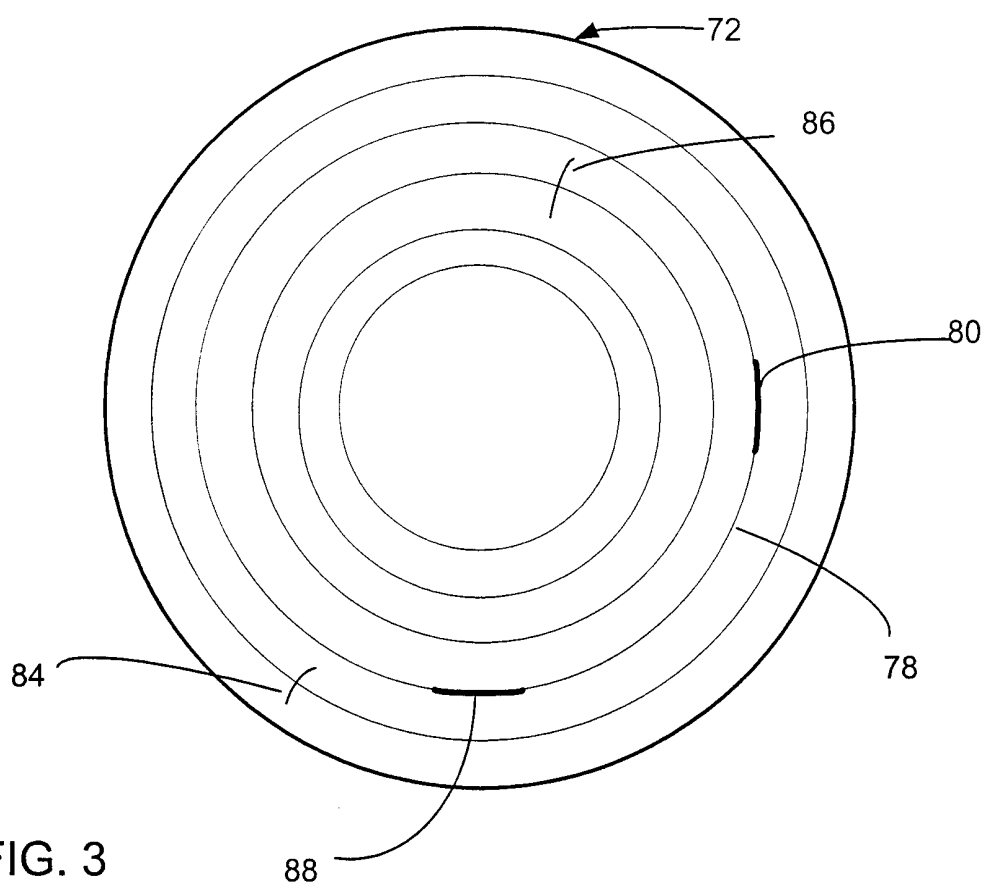
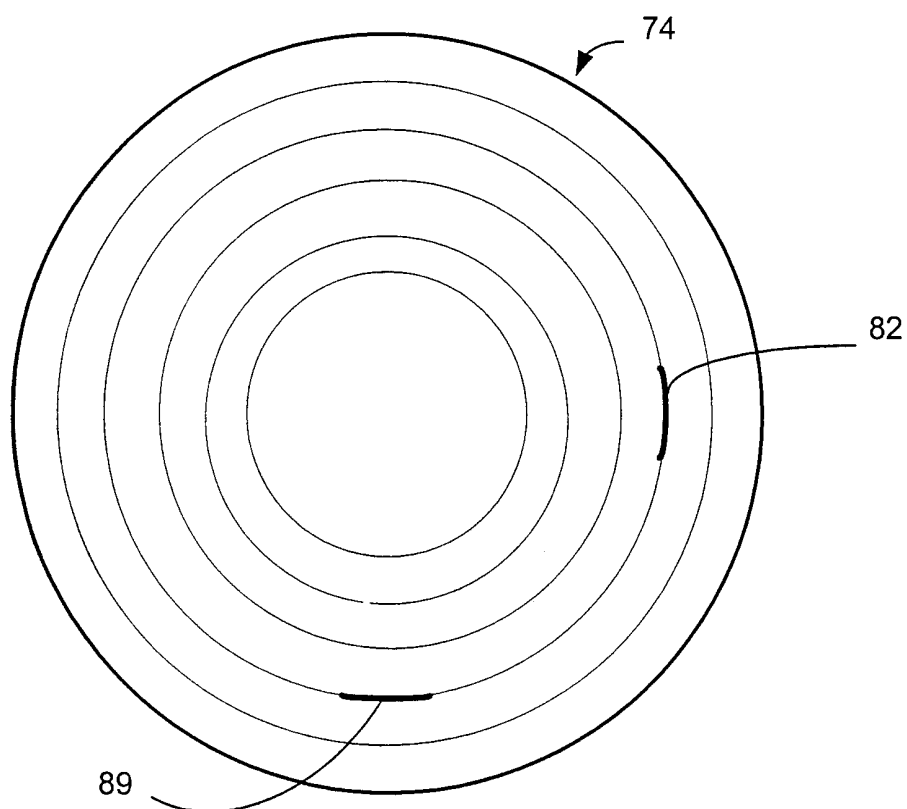


FIG. 3



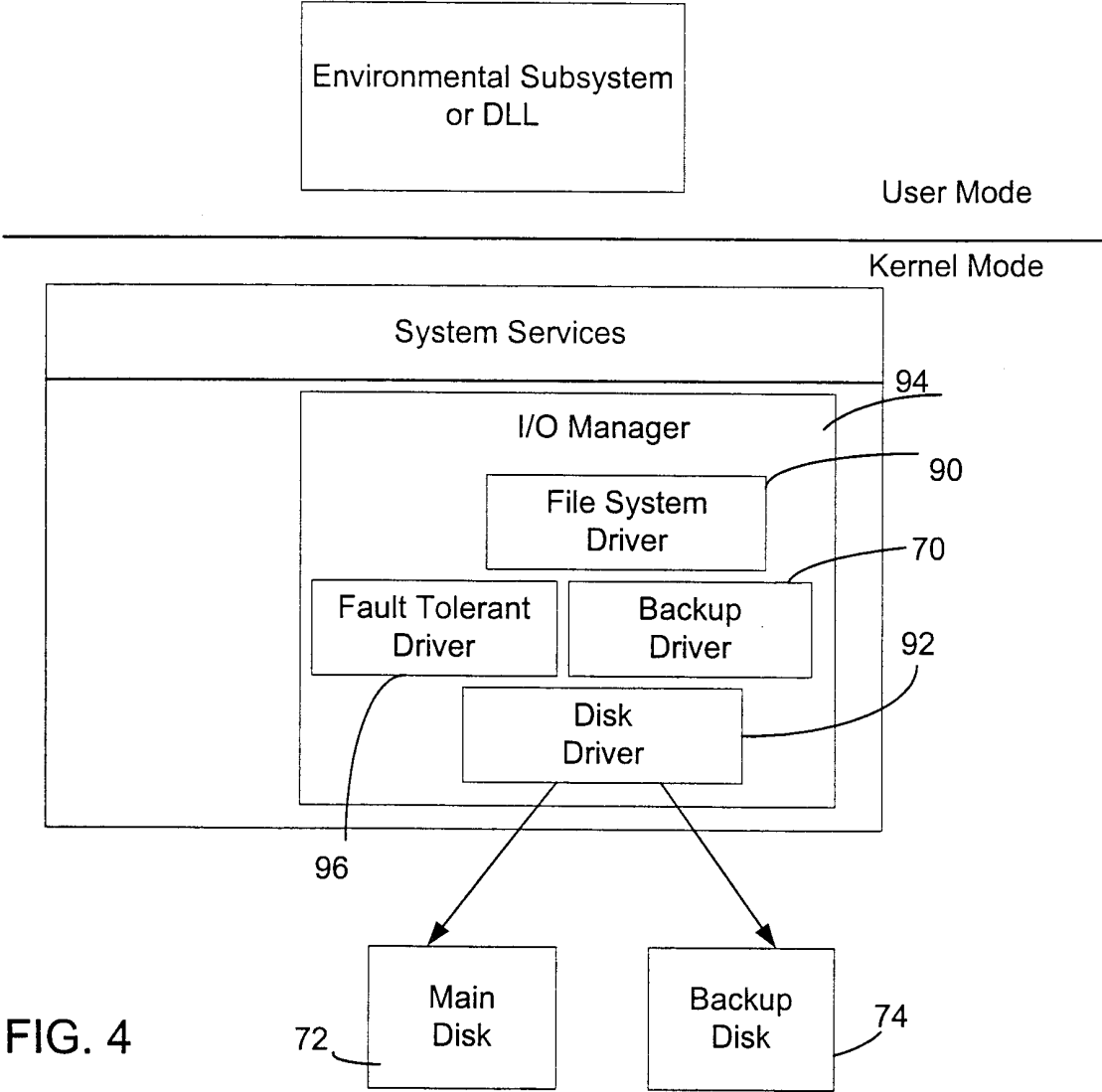


FIG. 4

5/5

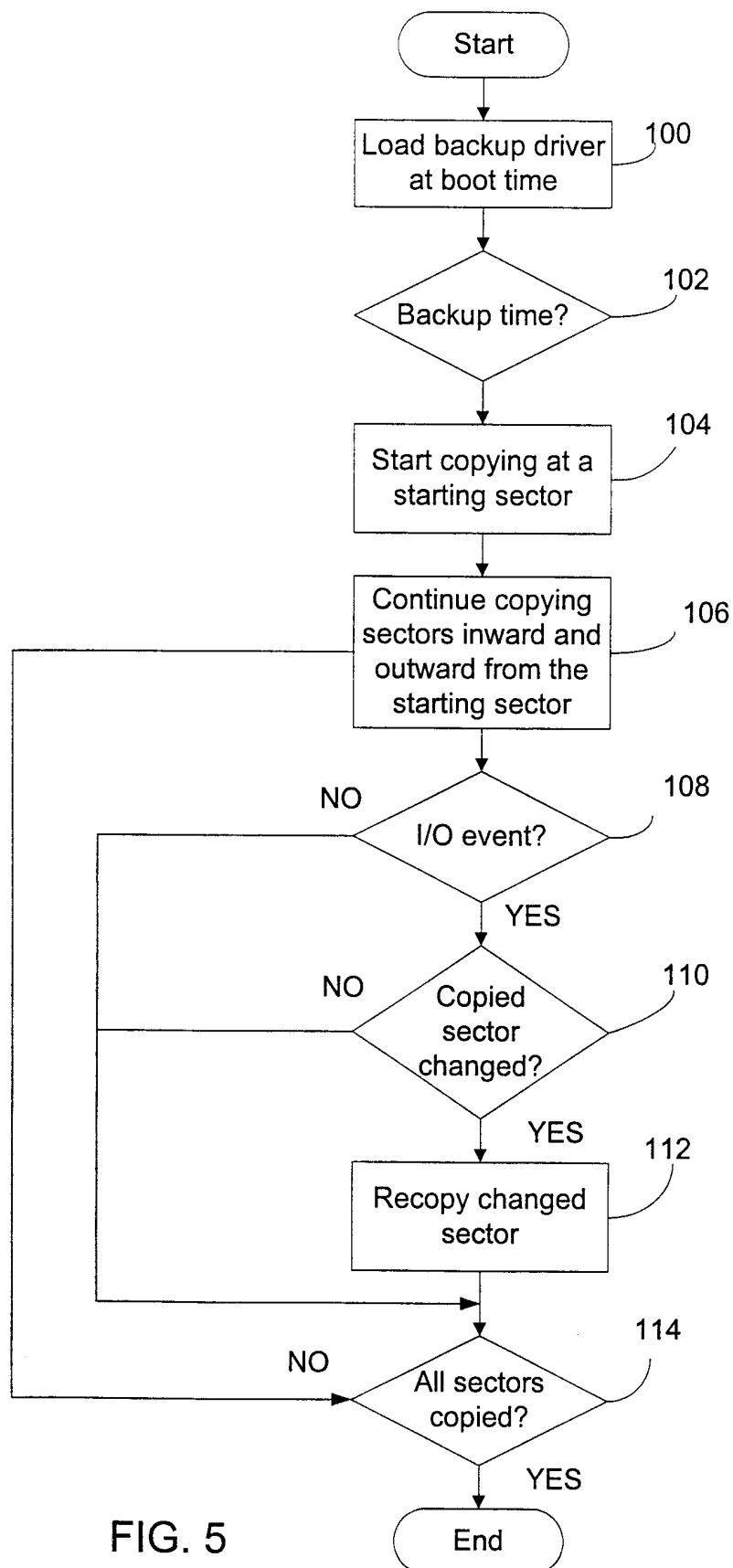


FIG. 5

INTERNATIONAL SEARCH REPORT

Int l Application No

PCT/US 00/16413

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F11/14

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)

IPC 7 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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	"THREE CHALLENGES IN THE IMPLEMENTATION OF DISK MIRRORING" IBM TECHNICAL DISCLOSURE BULLETIN, US, IBM CORP. NEW YORK, vol. 33, no. 4, 1 September 1990 (1990-09-01), pages 14-20, XP000124464 ISSN: 0018-8689	1-3, 5, 8, 10
Y A	page 14, line 1 -page 17, line 17; figures 1-4 --- -/--	7 4, 6, 9



Further documents are listed in the continuation of box C.



Patent family members are listed in 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.

& document member of the same patent family

Date of the actual completion of the international search

30 October 2000

Date of mailing of the international search report

07/11/2000

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

Fernandez Balseiro, J

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 00/16413

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	EP 0 767 431 A (STAC INC) 9 April 1997 (1997-04-09) column 8, line 23 - line 33	7
A	column 15, line 16 -column 16, line 37 ----	1-3,5,8, 10
A	US 5 487 160 A (BEMIS EDWARD D) 23 January 1996 (1996-01-23) column 3, line 22 - line 59 -----	1,8,10

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 00/16413

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0767431 A	09-04-1997	US 5907672 A JP 10055298 A	25-05-1999 24-02-1998
US 5487160 A	23-01-1996	NONE	