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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 May 2010 (06.05.2010)

PCT

(10) International Publication Number
WO 2010/049314 A1

(51) International Patent Classification:
G06F 11/14 (2006.01)

(21) International Application Number:
PCT/EP2009/063755

(22) International Filing Date:
20 October 2009 (20.10.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
08167951.6 30 October 2008 (30.10.2008) EP

(71) Applicant *(for all designated States except US)*: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **AGOMBAR, John, Paul** [GB/GB]; IBM United Kingdom Limited, MP 148, Husley Park, Hursey, Winchester Hampshire SO21 2JN (GB). **BEEKEN, Christopher, Barry** [GB/GB]; IBM United Kingdom Limited, MP 148, Hursley Park, Hursley, Winchester Hampshire SO21 2JN (GB). **FUENTE, Carlos, Francisco** [GB/GB]; IBM United Kingdom Limited, MP 148, Hursley Park, Hursley, Winchester Hampshire SO21 2JN (GB). **SCALES, William, James** [GB/GB]; IBM United Kingdom Limited, MP 148, Hursley Park, Hursley, Winchester Hampshire SO21 2JN (GB).

(74) Agent: **LING, Christopher, John**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester Hampshire SO21 2JN (GB).

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

[Continued on next page]

(54) Title: FLASHCOPY HANDLING

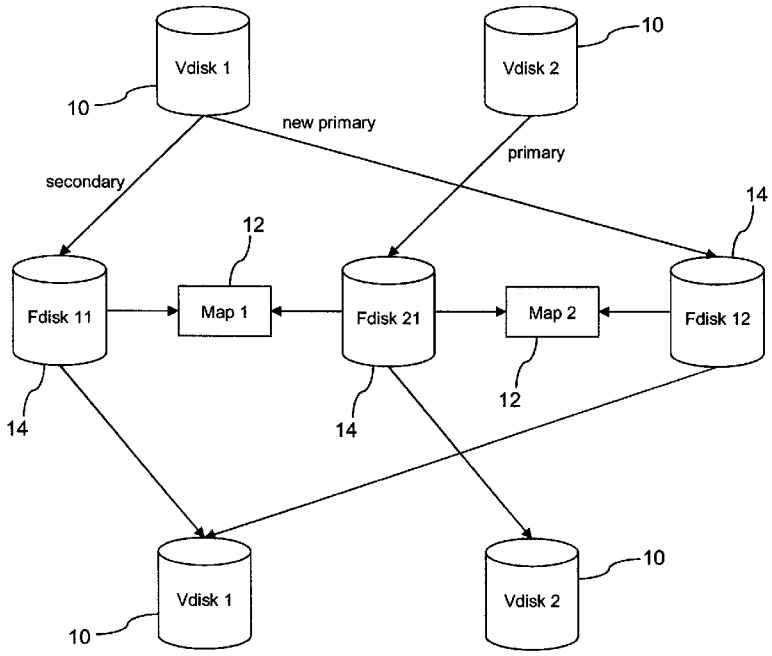


Fig. 4

(57) Abstract: A method of handling a flashcopy process comprises receiving a flashcopy instruction for a source disk, performing a flashcopy of the source disk on to a target disk, creating a map specifying the flashcopy from the source disk to the target disk, creating a primary fdisk for the source disk, if one does not already exist, and creating a primary fdisk for the target disk, if one does not already exist, or, if one does already exist, converting the existing primary fdisk for the target disk into a secondary fdisk, and creating a new primary fdisk for the target disk.

WO 2010/049314 A1

(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, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

— *with amended claims and statement (Art. 19(1))*

FLASHCOPY HANDLING

This invention relates to a method of, and system for, handling a flashcopy process. In one embodiment the method can be used for reversing a flashcopy mapping in a cascade.

A flashcopy is a feature supported on various storage devices that allows a user or an automated process to make nearly instantaneous copies of entire logical volumes of data. A copy of a source disk is made on a target disk. The copies are immediately available for both read and write access. A common feature of flashcopy like implementations is the ability to reverse the copy. That is, to populate the source disk of a flashcopy map with the contents of the target disk. With cascaded implementations, in which a target disk later becomes the source disk for a further flashcopy, this feature is complicated by the fact that the "grains" of data presented by a disk are located on a number of target/source disks "upstream" of the disk itself. This means that in order for a disk to maintain the image it is presenting, it must have access to all the disks containing these grains of data.

Reversing a flashcopy is the process of ensuring that the source disk presents the same image as the target disk, at the point in time that the reverse is started. This means that the contents of the source disk need to be modified in order to maintain the correct image. Now, if all of the target disks between the source and target disks in the cascade and the target disk itself are not needed, then the reverse can be achieved by simply taking offline all the target disks and reversing the order of the source and target disks in the cascade. Although this solution is sufficient for some use cases, it drastically restricts the flexibility of the flashcopy implementation, because in most cases the user will be reluctant to accept the loss of their intermediate flashcopy images, since they may contain other valid backup copies of the original source.

It is therefore an object of the invention to improve upon the known art.

According to a first aspect of the present invention, there is provided a method of handling a flashcopy process comprising receiving a flashcopy instruction for a source disk, performing a flashcopy of the source disk on to a target disk, creating a map specifying the flashcopy

from the source disk to the target disk, creating a primary fdisk for the source disk, if one does not already exist, and creating a primary fdisk for the target disk, if one does not already exist, or, if one does already exist, converting the existing primary fdisk for the target disk into a secondary fdisk, and creating a new primary fdisk for the target disk.

According to a second aspect of the present invention, there is provided a system for handling a flashcopy process comprising a user interface arranged to receive a flashcopy instruction for a source disk, and a processor arranged to perform a flashcopy of the source disk on to a target disk, to create a map specifying the flashcopy from the source disk to the target disk, to create a primary fdisk for the source disk, if one does not already exist, and to create a primary fdisk for the target disk, if one does not already exist, or, if one does already exist, to convert the existing primary fdisk for the target disk into a secondary fdisk, and to create a new primary fdisk for the target disk.

According to a third aspect of the present invention, there is provided a computer program product on a computer readable medium for handling a flashcopy process, the product comprising instructions for receiving a flashcopy instruction for a source disk, performing a flashcopy of the source disk on to a target disk, creating a map specifying the flashcopy from the source disk to the target disk, creating a primary fdisk for the source disk, if one does not already exist, and creating a primary fdisk for the target disk, if one does not already exist, or, if one does already exist, converting the existing primary fdisk for the target disk into a secondary fdisk, and creating a new primary fdisk for the target disk.

Owing to the invention, it is possible to provide a data structure that will support the use of cascaded flashcopies using multiple source and target disks that will support reversal of any flashcopy back to a source disk, without losing any of the flashcopy images that are represented within the cascade. Standard IO commands can still be applied to any of the source or target disks, and the nature of the data structure that has been created supports a simple handling of the IO commands, which lead to the flashcopies being correctly processed through the cascade.

The proposed solution is to allow a disk to exist a number of times in the same cascade or in separate cascades. In order to reverse a flashcopy map a user can simply create and start a map in the opposite direction. For example, a map from VDisk1 to VDisk2 can be reversed by creating a new map from VDisk2 to VDisk1. The present invention describes a scheme that allows both maps to be active simultaneously. This means that the image presented by VDisk2 can be maintained at the same time as VDisk1 is recovered.

This scheme introduces the concept of an fdisk, which is internal to flashcopy, and is used to represent the source or target disk for a given flashcopy map in the cascade. The flashcopy cascades are constructed between maps and fdisks. The scheme enables a map to be reversed by allowing a VDisk to be represented in multiple cascades or in multiple points in the same cascade by different fdisks. One fdisk will represent the VDisk that is being presented to the host and the other fdisks will represent the VDisk required to maintain the cascade at other points. An fdisk, in one embodiment, is an aspect of a VDisk. In this implementation, the fdisk could for example by an object (structure) which links the VDisk to the maps upstream and downstream of the position within the cascade.

Preferably, the method further comprising receiving an IO command for a specific disk, and processing the IO command according to a predefined set of rules relating to the existence of primary and secondary fdisks for the specific disk. In one embodiment, the predefined set of rules define cleaning of a secondary fdisk prior to processing the IO command, the cleaning comprising copying data from the specific disk to the target disk defined in the map linked to the secondary disk. The data structure can be used to maintain the correct structures in the cascade, when IO commands are received by disks, wherever they are in the cascade. The rules define the handling of the IO instructions.

Embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:-

Figure 1 is a schematic diagram of a pair of storage disks,

Figure 2 is a schematic diagram of a cascade of storage disks,

Figure 3 is a schematic diagram of a data structure created following a flashcopy,

Figure 4 is a schematic diagram of a further data structure, and

Figure 5 is a flowchart of a method of handling a flashcopy.

Figure 1 illustrates the concept of a flashcopy using two disks 10. The disks 10 could form part of a larger array of disks, and would typically form part of an enterprise storage solution. The disks 10 could be part of a storage solution relating to a commercial website, for example. If at any time a backup needs to be made of the content of Vdisk1, then a flashcopy instruction can be sent to that disk, which defines a source disk (Vdisk1) and also a target disk, which is the target of the flashcopy. The flashcopy instruction creates a point in time copy of the image of the disk which is the source disk.

In the example of Figure 1, the source disk of a first flashcopy instruction is Vdisk1, and the target disk is Vdisk2. The flashcopy instruction starts the flashcopy process, which creates a map 12 from the source disk 10 to the target disk 10. This first map is labelled map 1 in the Figure. The image of Vdisk1 at this specific point in time is now available on Vdisk2. This creates a backup of the data on Vdisk1, and also allows tests and other administration tasks to be run on the data of Vdisk1, without the attendant danger of losing any of the original data, as it is preserved on the original source disk.

When a flashcopy is made, it creates a link between the two disks 10, as defined by the map 12. Data may now be copied across in the background, with the additional requirement that any access to Vdisk2 (as the target disk 10) will immediately cause the relevant parts of the image of Vdisk1 to be copied across, and also any access to Vdisk1 which would result in a change to the image stored by that disk 10 will also cause the unaltered data to be immediately copied across. In this way, the Vdisk2, to an outside user, stores the point in time copy of Vdisk1, although data will only be physically copied across under the circumstances described above.

In order to reverse a flashcopy map, the user or administrator can simply create and start a map 12 in the opposite direction. The user issues a flashcopy instruction that specifies a map from a new source to target disk. For example, the map 12 from VDisk1 to VDisk2 can be reversed by creating a new map (map 2) from VDisk2 to VDisk1. The system of the present invention, described in more detail below, describes a scheme that allows both maps 12 to be active simultaneously. This means that the image presented by VDisk2 can be maintained at the same time as VDisk1 is recovered. This is valuable in relation to, for example, flashcopy cascades.

Figure 2 illustrates a flashcopy cascade, which exists in relation to four disks 10 that form part of an enterprise storage solution. Each map 12 within the Figure represents a flashcopy instruction that has resulted in a point in time copy of a Vdisk 10 being taken and passed from a source disk 10 to a target disk 10. A user has at various times requested that a disk 10 that has been the target for an earlier flashcopy, should become the source for a later flashcopy, in a process that creates a cascade of flashcopies. Actions can be taken in respect of an image on a Vdisk 10 after a flashcopy has been made.

The system allows a disk 10 to exist a number of times in the same cascade or in separate cascades. The scheme of the invention will extend to more complex graphs of trees for example. In this example, before map 4 is started, VDisk1 contains the users original set of data and VDisk2 contains a backup of that data. VDisk3 and VDisk4 contain further snapshots of the data on which two different simulations have been run. The user then decides that they like the result of the simulation on VDisk3 and want to restore this data back onto VDisk1. The scheme allows the user to start map 4 and copy the modified image from VDisk3 to Vdisk1 whilst still maintaining the unmodified backup of the original data on VDisk2.

This scheme operated by a system according to the invention, introduces the concept of an fdisk, which is a data structure internal to flashcopy, and is used to represent the source or target disk 10 for a given flashcopy map in the cascade. The flashcopy cascades are then constructed between maps 12 and fdisks. The scheme enables a map 12 to be reversed by allowing a VDisk 10 to be represented in multiple cascades or in multiple points in the same

cascade by different fdisks. A first fdisk will represent the VDisk that is being presented to the host and the other fdisks will represent the VDisk required to maintain the cascade at other points.

Figure 3 illustrates the data structure created by the system, in relation to the disks of Figure 1, after map1 has been formed. The fdisks 14 are created, whenever there is a flashcopy instruction. The fdisk 14 is a logical structure, rather than the physical structure of the source and target disks 10. In the first example above, there is a map Map 1 from VDisk1 to VDisk2. When this map 12 is started, the VDisk and fdisk are arranged so that IO (input/output) commands directed to VDisk1 are redirected to FDisk11 and IO commands to VDisk2 are redirected to FDisk21. The fdisks 14 will process IO commands according to the usual cascaded algorithm. IO from the fdisk is then redirected to the appropriate VDisk below.

In the simple structure of Figure 1, after the flashcopy instruction, which creates the first map 1, the creation of the fdisks 14 is a basis for dealing with more complicated cascading flashcopies. Each fdisk 14 that is created can be either a primary fdisk 14 or a secondary fdisk 14. The scheme that is used in relation to the creation of the fdisks 14 for the disks 10 is that there is created a primary fdisk 14 for the source disk 10, if one does not already exist. Additionally, there is created a primary fdisk 14 for the target disk 10, if one does not already exist, or, if one does already exist, the existing primary fdisk 14 for the target disk 10 is converted into a secondary fdisk 14, and a new primary fdisk 14 is created for the target disk 10. This data structure supports the handling of the cascading flashcopies.

For example, in Figure 1, after the creation of the flashcopy backup with map 1, the user reversed the process with the flashcopy which is embodied in map 2. The data structure of Figure 3 would be amended after this action, to look like the data structure of Figure 4. In order to reverse VDisk2 onto VDisk1 the user creates map 2 from VDisk2 to VDisk1. When map 2 is started, a new primary fdisk 14 is created for VDisk1 and the current primary fdisk 14 is demoted to a secondary role. This is shown in the Figure, where new FDisk12 has been created, which is linked to VDisk1, and the pre-existing FDisk11 is changed to a secondary fdisk.

When future IO commands are directed to VDisk1, then firstly the secondary fdisk data must be "cleaned" and then the IO command is directed to FDisk12 as normal. Cleaning of an fdisk involves copying the existing data from that source disk onto the target disk of the first dependent map in the cascade. In this example, this would involve copying the data from Vdisk1 to Vdisk2. In this way the image presented by VDisk2 can be maintained. All future IO commands are handled according to a predefined set of rules, which relate to the existence of primary and secondary fdisks for the specific disk to which the specific IO command relates.

In the structure of Figure 4 the fdisks could be objects defined by:

FDisk 11:

Vdisk Index: 1;
Up Map: NULL;
Dn Map: Map1

FDisk 12:

Vdisk Index: 1;
Up Map: Map 2
Dn Map: NULL

FDisk 21:

Vdisk Index: 2;
Up Map: Map 1
Dn Map: Map 2

Each fdisk has an index defining the vdisk to which it relates, and has links to the relevant maps that define the up and down directions of the original flashcopies in the cascade.

In general, each VDisk has a primary fdisk and may have a secondary fdisks. A VDisk can be in one of four states governed by its primary and secondary fdisks. These states are no primary, no secondary; active primary, no secondary; active primary, active secondary; and no primary, active secondary. If there is no primary and no secondary then this implies that no flashcopy cascade is interested in the specific VDisk and so all IO will be passthrough. If there is an active primary, but no secondary, then this implies that the IO commands for this VDisk should be passed on to the primary fdisk for processing. There is no pre-IO cleaning to be carried out. If there is an active primary and an active secondary then this implies that the IO for this VDisk should be passed on to the primary fdisk for processing. Prior to any processing of the IO, the secondary should first be cleaned. Finally, if there is no primary, but an active secondary, then before processing the IO, the secondary should first be cleaned. Once the secondary is cleaned the IO is passthrough.

The above has been described to include the state of "no primary, active secondary". Whether a VDisk can actually exist without a primary is dependent on the implementation. For instance, an implementation may allow the user to prematurely stop a map. This would destroy the Target fdisk, which maybe a primary. At this point, the implementation could promote the secondary to a primary, or it could leave the Vdisk with no primary. It is for the implementers to decide which approach makes the most sense, but both would work.

The method of handling the flashcopy process to create the data structure is summarised in Figure 5. The method comprises, firstly, step S1, which is the step of receiving a flashcopy instruction for a source disk. This may be received from an administrator through a suitable user interface. The remainder of the method is carried out by a processing function that is part of, or attached to, the enterprise storage system that comprises the various disks.

The first step S1 is followed by step S2, in which there is performed the flashcopy of the source disk on to a target disk. The next step is step S3, which comprises creating a map specifying the flashcopy from the source disk to the target disk. In this way a flashcopy is carried out and a map is created that contains the detail of the specific flashcopy. Note that the flashcopy could be a reversal of an earlier flashcopy, but both actions are considered to be flashcopies in the sense of copying data from a source disk to a target disk.

There is next carried out a step to check is there exists a primary fdisk for the source disk. If not, then method proceeds to step S4, which comprises creating a primary fdisk for the source disk, as one does not already exist. Note that if an fdisk already exists for the source disk, there is no need to take any action in respect of that source disk, and the method simply moves to the next check.

Once the creation of an fdisk has been completed for the source disk, if one was needed, then the next stage is to check whether a primary fdisk exists for the target disk. If the answer is no, then at step S5, there is created a primary fdisk for the target disk, as one does not already exist. If one does already exist, then the method proceeds to step S6, which comprises converting the existing primary fdisk for the target disk into a secondary fdisk. The final step of the method is the step S7 of creating a new primary fdisk for the target disk.

In this way, the data structure of fdisks and maps is created each time a flashcopy is made of any source disk. Preferably the step S3 of the method also comprises linking the created map to each created primary fdisk. This ensures that a data structure is present that will support the cascading of flashcopies through disks that nevertheless provides sufficient flexibility to obtain the data at different places within the cascade, without requiring the need for any complicated administrator intervention.

The scheme will also allow for any number of secondary fdisks. The number of fdisks associated with a disk determines the rules for which maps can be started simultaneously. A balance between flexibility and performance needs to be set because of the extra workload due to the cleaning IO redirected to the secondaries that must be completed before the primary IO can proceed.

As discussed above, once an IO command has been received for a specific disk, then the processing of the IO command is according to a predefined set of rules relating to the existence of primary and secondary fdisks for the specific disk. Advantageously, the predefined set of rules define cleaning of a secondary fdisk prior to processing the IO

command, the cleaning comprising copying data from the specific disk to the target disk defined in the map linked to the secondary disk.

CLAIMS

1. A method of handling a flashcopy process comprising:

- receiving a flashcopy instruction for a source disk,
- performing a flashcopy of the source disk on to a target disk,
- creating a map specifying the flashcopy from the source disk to the target disk,
- creating a primary fdisk for the source disk, if one does not already exist, and
- creating a primary fdisk for the target disk, if one does not already exist, or, if one does already exist, converting the existing primary fdisk for the target disk into a secondary fdisk, and creating a new primary fdisk for the target disk.

2. A method according to claim 1, and further comprising linking the map to the or each created primary fdisk.

3. A method according to claim 1 or 2, and further comprising receiving an IO command for a specific disk, and processing the IO command according to a predefined set of rules relating to the existence of primary and secondary fdisks for the specific disk.

4. A method according to claim 2 and 3, wherein the predefined set of rules define cleaning of a secondary fdisk prior to processing the IO command, the cleaning comprising copying data from the specific disk to the target disk defined in the map linked to the secondary disk.

5. A system for handling a flashcopy process comprising:

- a user interface arranged to receive a flashcopy instruction for a source disk, and
- a processor arranged to perform a flashcopy of the source disk on to a target disk, to create a map specifying the flashcopy from the source disk to the target disk, to create a primary fdisk for the source disk, if one does not already exist, and to create a primary fdisk for the target disk, if one does not already exist, or, if one does already exist, to convert the existing primary fdisk for the target disk into a secondary fdisk, and to create a new primary fdisk for the target disk.

6. A system according to claim 5, wherein the processor is further arranged to link the map to the or each created primary fdisk.

7. A system according to claim 5 or 6, wherein the processor is further arranged to receive an IO command for a specific disk, and to process the IO command according to a predefined set of rules relating to the existence of primary and secondary fdisks for the specific disk.

8. A system according to claim 6 and 7, wherein the predefined set of rules define cleaning of a secondary fdisk prior to processing the IO command, the cleaning comprising copying data from the specific disk to the target disk defined in the map linked to the secondary disk.

9. A computer program product on a computer readable medium for handling a flashcopy process, the product comprising instructions for:

- receiving a flashcopy instruction for a source disk,
- performing a flashcopy of the source disk on to a target disk,
- creating a map specifying the flashcopy from the source disk to the target disk,
- creating a primary fdisk for the source disk, if one does not already exist, and
- creating a primary fdisk for the target disk, if one does not already exist, or, if one does already exist, converting the existing primary fdisk for the target disk into a secondary fdisk, and creating a new primary fdisk for the target disk.

10. A computer program product according to claim 9, and further comprising instructions for linking the map to the or each created primary fdisk.

11. A computer program product according to claim 9 or 10, and further comprising instructions for receiving an IO command for a specific disk, and processing the IO command according to a predefined set of rules relating to the existence of primary and secondary fdisks for the specific disk.

12. A computer program product according to claim 10 and 11, wherein the predefined set of rules define cleaning of a secondary fdisk prior to processing the IO command, the cleaning comprising copying data from the specific disk to the target disk defined in the map linked to the secondary disk.

AMENDED CLAIMS

received by the International Bureau on 07 April 2010 (07.04.2010)

1. A method of handling a flashcopy process comprising:

- receiving a flashcopy point-in-time copy instruction for a source disk,
- performing a flashcopy point-in-time copy of the source disk on to a target disk,
- creating a flashcopy map specifying the flashcopy data population relationship from the source disk to the target disk,
- creating a primary fdisk source disk representation for the source disk, if one does not already exist, and
- creating a primary fdisk target disk representation for the target disk, if one does not already exist, or, if one does already exist, converting the existing primary fdisk for the target disk into a secondary fdisk, and creating a new primary fdisk for the target disk.

2. A method according to claim 1, and further comprising linking the flashcopy map to the or each created primary fdisk.

3. A method according to claim 1 or 2, and further comprising receiving an IO command for a specific disk, and processing the IO command according to a predefined set of rules relating to the existence of primary and secondary fdisks for the specific disk.

4. A method according to claim 2 and 3, wherein the predefined set of rules define cleaning of a secondary fdisk prior to processing the IO command, the cleaning comprising copying data from the specific disk to the target disk defined in the flashcopy map linked to the secondary disk.

5. A system for handling a flashcopy point-in-time copy process comprising:

- a user interface arranged to receive a flashcopy point-in-time copy instruction for a source disk, and
- a processor arranged to perform a flashcopy point-in-time copy of the source disk on to a target disk, to create a flashcopy map specifying the flashcopy data population relationship from the source disk to the target disk, to create a primary fdisk source disk representation for the source disk, if one does not already exist, and to create a primary fdisk

target disk representation for the target disk, if one does not already exist, or, if one does already exist, to convert the existing primary fdisk for the target disk into a secondary fdisk, and to create a new primary fdisk for the target disk.

6. A system according to claim 5, wherein the processor is further arranged to link the flashcopy map to the or each created primary fdisk.

7. A system according to claim 5 or 6, wherein the processor is further arranged to receive an IO command for a specific disk, and to process the IO command according to a predefined set of rules relating to the existence of primary and secondary fdisks for the specific disk.

8. A system according to claim 6 and 7, wherein the predefined set of rules define cleaning of a secondary fdisk prior to processing the IO command, the cleaning comprising copying data from the specific disk to the target disk defined in the flashcopy map linked to the secondary disk.

9. A computer program comprising computer program code to, when loaded into a computer system and executed thereon, cause said computer system to perform all the steps of the method according to any one of claims 1 to 4.

STATEMENT UNDER ARTICLE 19 PCT

The claims have been amended to address the clarity issues raised in the Written Opinion of the International Searching Authority and to conform independent claim 9 to the requirements of Article 33(3) PCT.

As, under Article 19 PCT, only the claims may be amended, the mention of Documents D1 and D2 has not been added to the description, nor has the summary of the invention been brought into conformity with the claims as amended. Formal matters relating to reference numerals and the two-part form of claiming are also not addressed in the amendments under Article 19 as there are variations in the requirements of the various jurisdictions in which protection may be sought during the National and Regional Phase. The applicant proposes to address all these matters in the National or Regional Phase according to the requirements of the various jurisdictions.

Yours faithfully



P J Stretton
European Patent Attorney
Authorised Representative
IBM United Kingdom Limited
PJS/hh

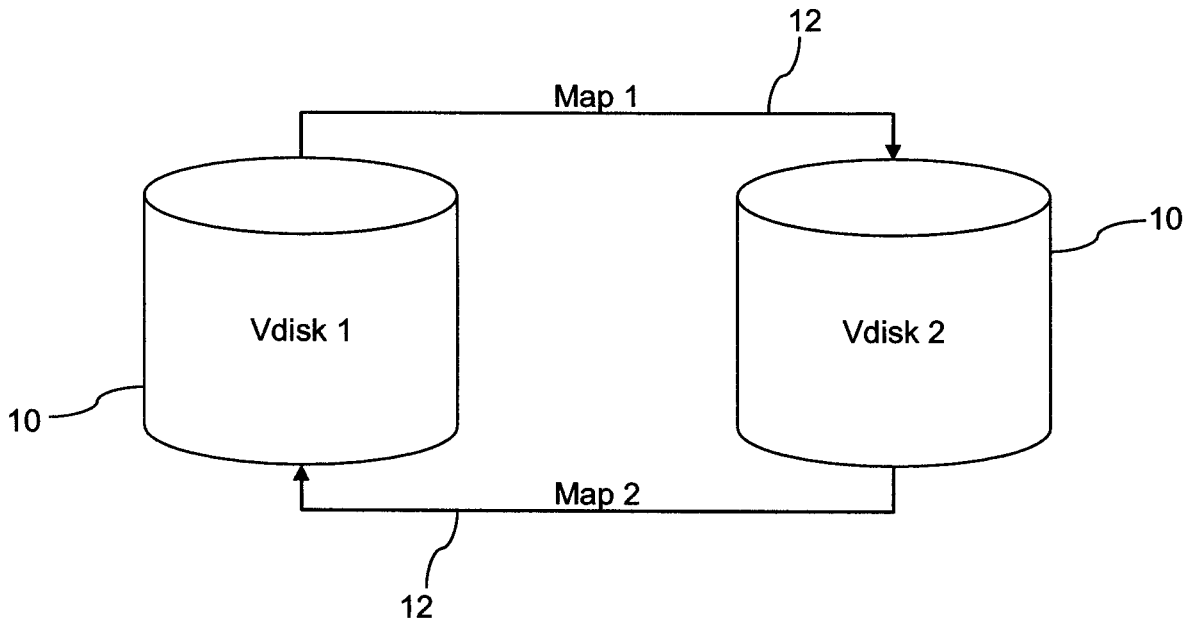


Fig. 1

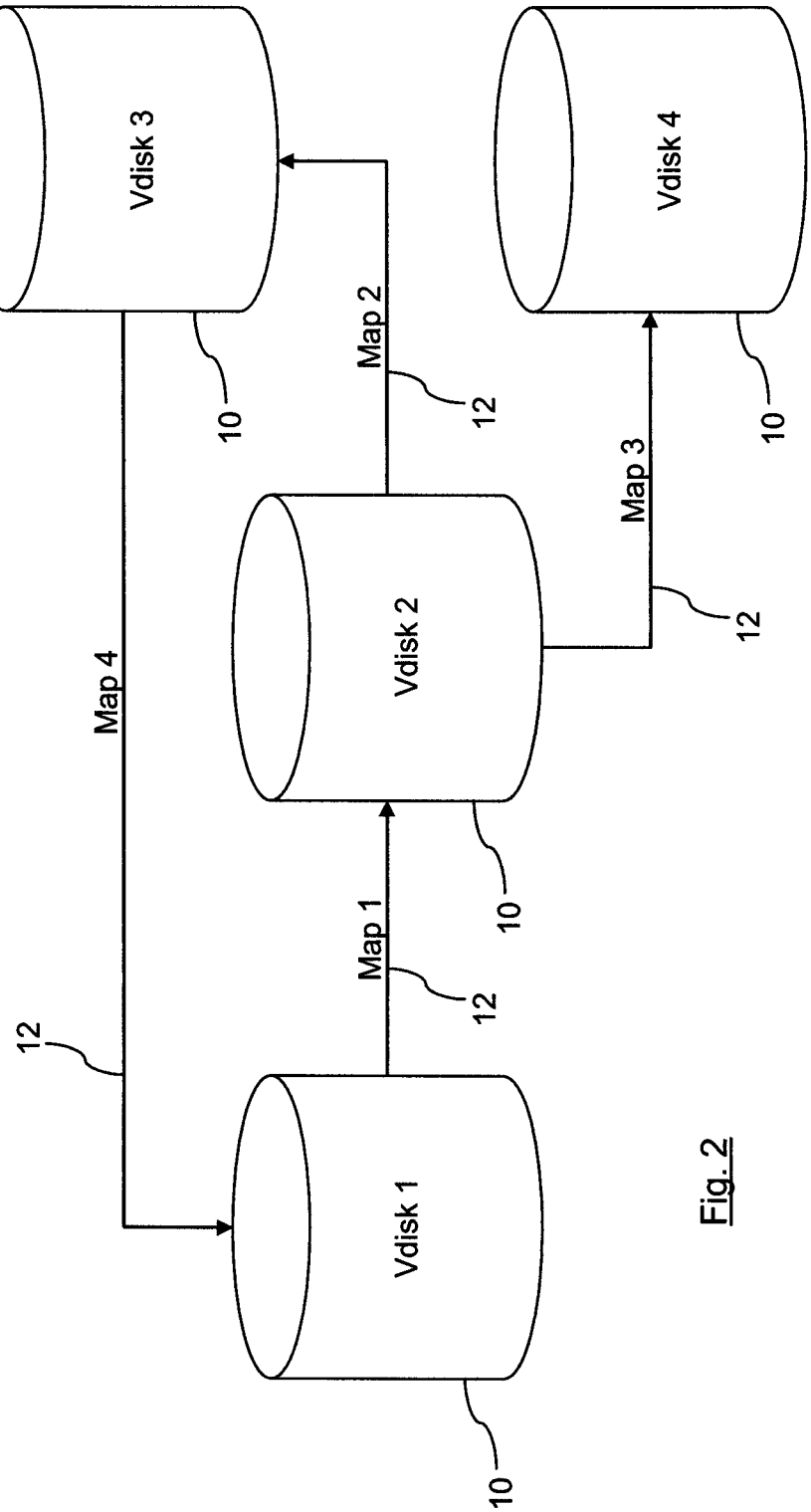


Fig. 2

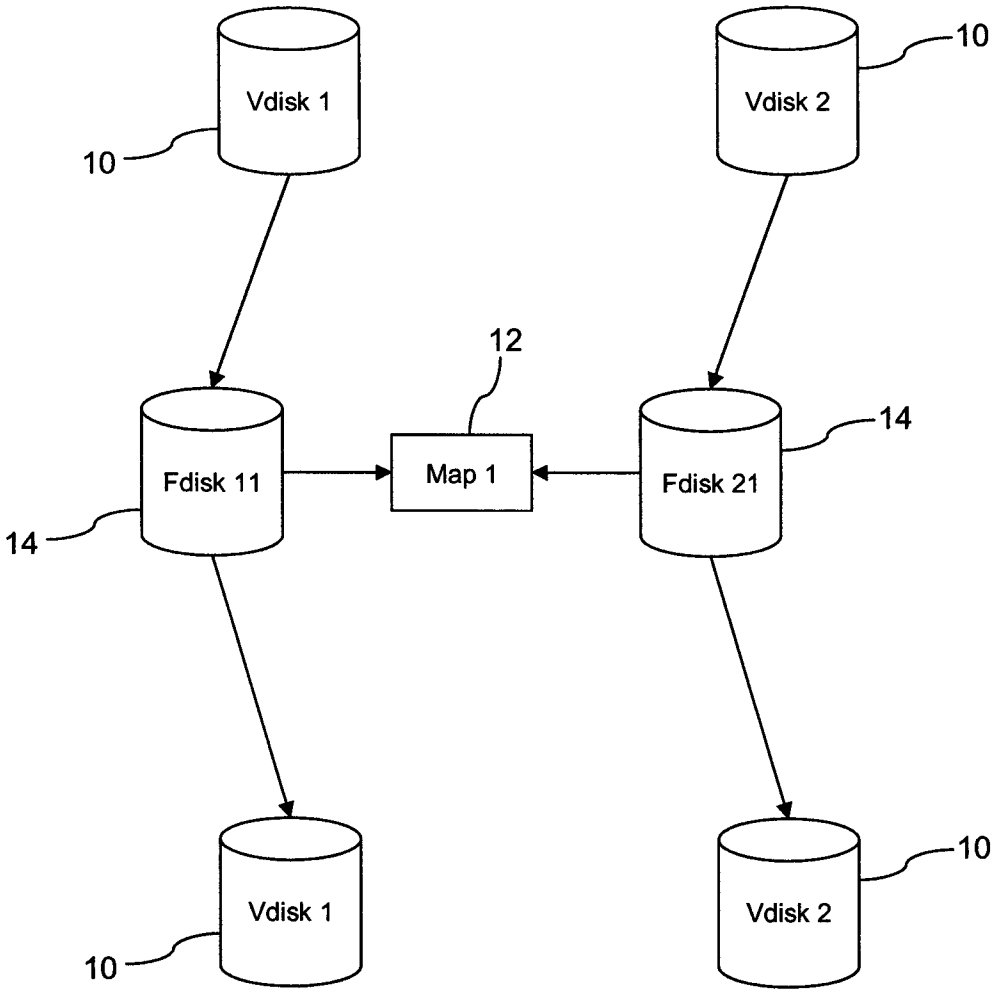

Vdisk 1
10
Vdisk 2
10
12
Fdisk 11
14
Map 1
Fdisk 21
14
Vdisk 1
10
Vdisk 2
10

Fig. 3

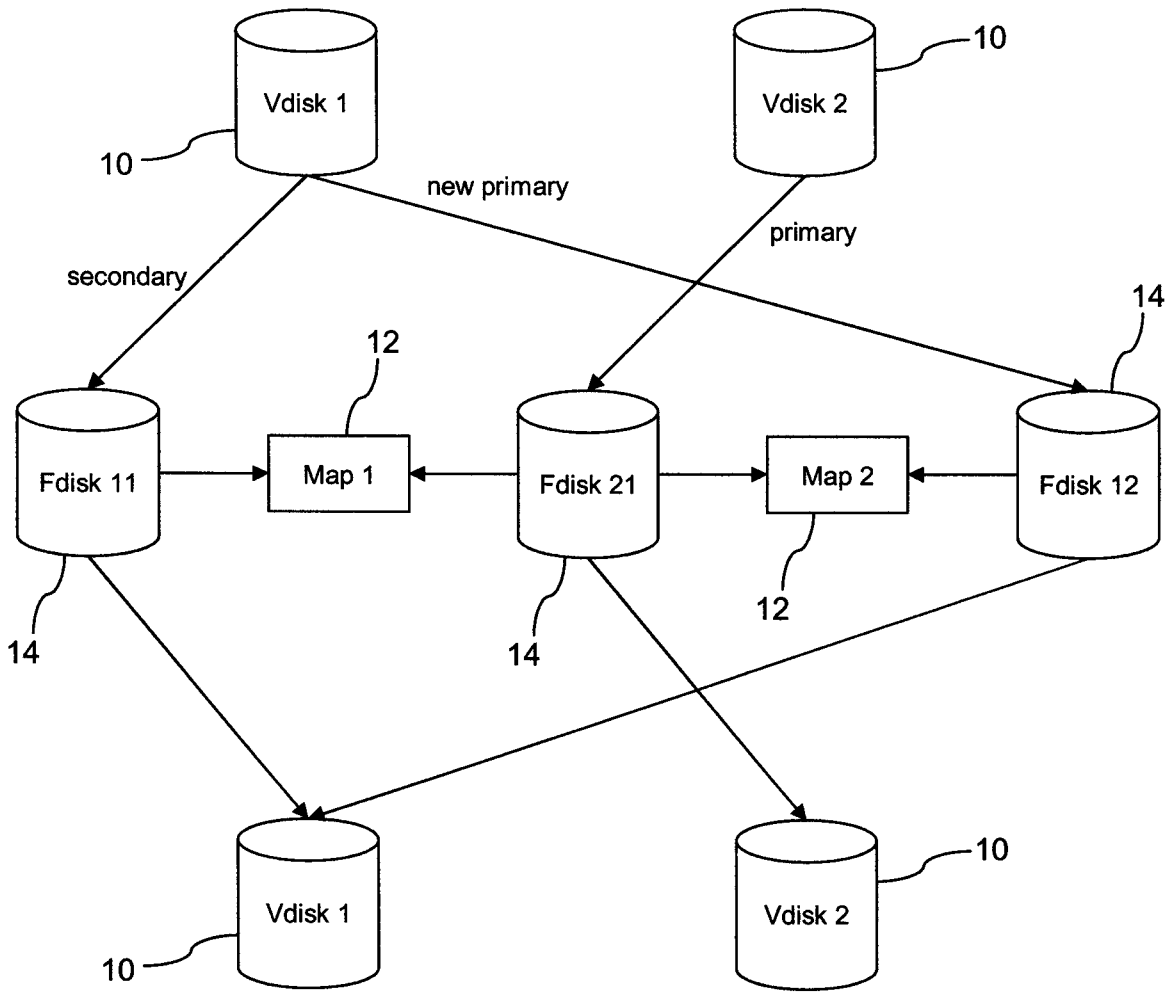

Vdisk 1
Vdisk 2
10
10
new primary
primary
secondary
14
12
Fdisk 11
Map 1
Fdisk 21
Map 2
Fdisk 12
12
14
14
Vdisk 1
Vdisk 2
10
10

Fig. 4

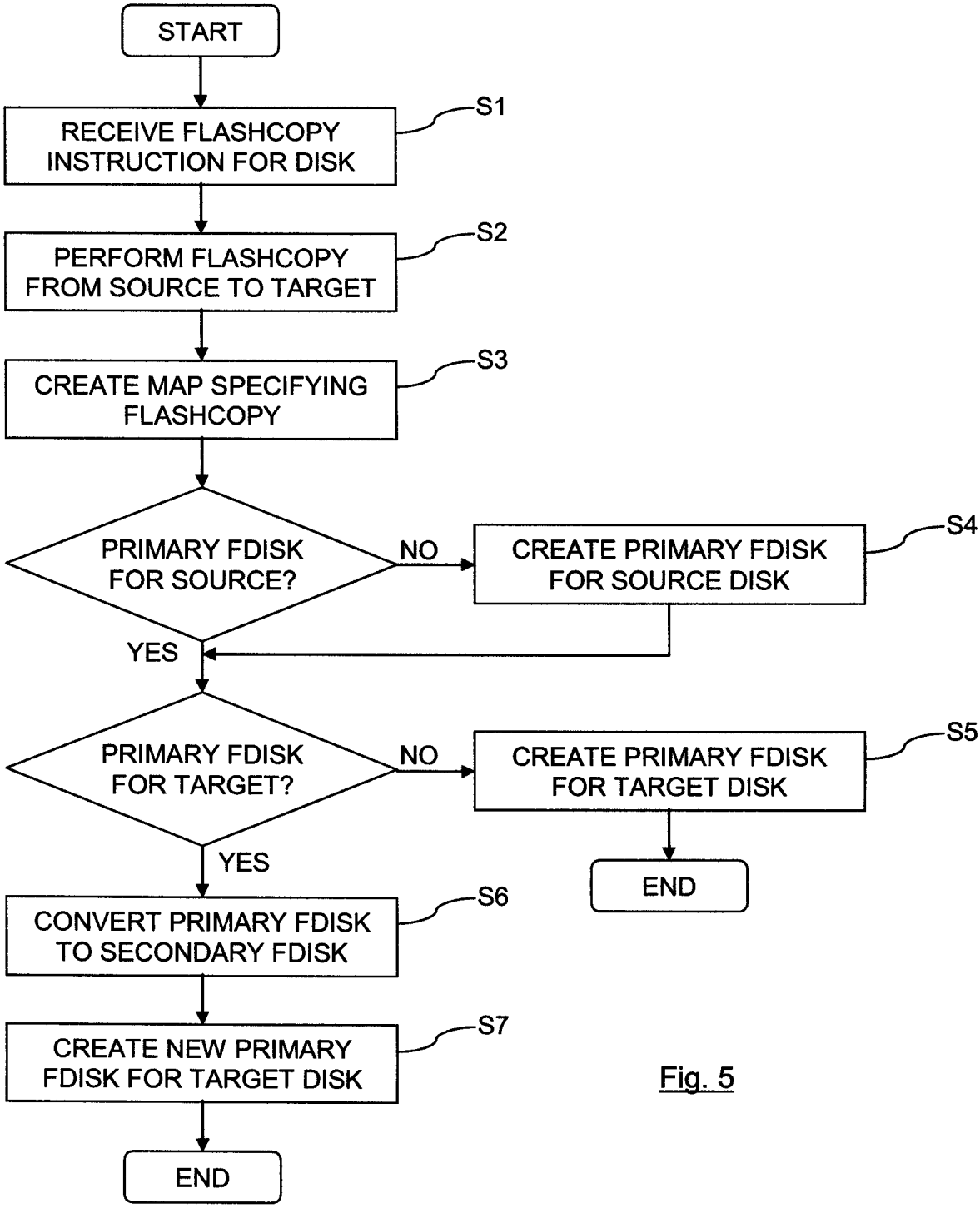

START
RECEIVE FLASHCOPY INSTRUCTION FOR DISK
S1
PERFORM FLASHCOPY FROM SOURCE TO TARGET
S2
CREATE MAP SPECIFYING FLASHCOPY
S3
PRIMARY FDISK FOR SOURCE?
NO
CREATE PRIMARY FDISK FOR SOURCE DISK
S4
YES
PRIMARY FDISK FOR TARGET?
NO
CREATE PRIMARY FDISK FOR TARGET DISK
S5
YES
END
CONVERT PRIMARY FDISK TO SECONDARY FDISK
S6
CREATE NEW PRIMARY FDISK FOR TARGET DISK
S7
END

Fig. 5

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2009/063755

A. CLASSIFICATION OF SUBJECT MATTER
INV. 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)
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, COMPENDEX, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2005/278391 A1 (SPEAR GAIL A [US] ET AL SPEAR GAIL ANDREA [US] ET AL) 15 December 2005 (2005-12-15) paragraphs [0012], [0013], [0015], [0016], [0018], [0019], [0032], [0036] - [0040], [0054], [0061] - [0071], [0076] - [0082] figures 1,5a,5b,8	1-12
A	US 2006/139697 A1 (FUENTE CARLOS F [GB]) 29 June 2006 (2006-06-29) paragraphs [0008] - [0010], [0026] - [0053] figures 2,3,4a,4b,4c,5,6	1-12

☐ 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.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
1 February 2010	09/02/2010

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Fax: (+31–70) 340–3016	Johansson, Ulf

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2009/063755

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2005278391 A1	15-12-2005	CN 1702627 A	30-11-2005
		JP 2005339554 A	08-12-2005
		US 2009187613 A1	23-07-2009
US 2006139697 A1	29-06-2006	CN 1794161 A	28-06-2006
		US 2008189482 A1	07-08-2008