

Method and Apparatus for Storing Data

Field of the Invention

5 The present invention relates to computer technologies, and more particularly to, a method and apparatus for storing data.

Background of the Invention

File systems of computers are core modules of file management for applications (such as word and Instance Managing applications etc.), and manage data of the applications and implement local persistence.

10 Compound Documents are one of good solutions for implementing the file system. By using the compound document, multiple files are organized in catalog levels, which include not only texts, but also graphs, spreadsheet data, voice, video images and other information. The compound document may be regarded as an owner including the texts, the graphs and multimedia information (such as voice and images), and the multiple files
15 are organized in catalog levels.

However, because each file is stored in catalog levels in compound document, when the file is being stored, if the application of the computer becomes abnormal (such as power-off, dead halt, process crash, deadlock and so on), it is very possible that data of the file system is damaged, and the damaged data always causes various problems of the
20 application.

Summary of the Invention

Embodiments of the present invention provide a method and apparatus for storing data, so that the file system can provide a highly reliable data storage system for the applications and provide transactional support for upper applications.

25 The technical solutions of the present invention include:

A method for storing data, including:

receiving operation data of an operation object in a transaction period, and buffering

the operation data in a memory;

storing a backup of original data of the operation object in a compound document in a log file;

storing the operation data buffered in the memory into the compound document.

5 An apparatus for storing data, including:

a receiving module, adapted to receive operation data of an operation object in a transaction period, and buffer the operation data in a memory;

a backup module, adapted to store a backup of original data of the operation object in a compound document in a log file;

10 a storing module, adapted to store the operation data buffered in the memory into the compound document after the backup module stores the backup of the original data of the operation object in the compound document in the log file.

The technical effects brought by the technical solutions of the embodiments of the present invention include the following.

15 In the embodiments of the present invention, the operation data is firstly buffered in the memory, and the backup of the original data of the operation object in the compound document is stored in the log file before the operation data is stored into the compound document. If no abnormal situation occurs in the procedure of storing the operation data buffered in the memory into the compound document of the operation object, the “All”
20 effect of the “All of Nothing” transaction features is implemented; If the abnormal situation occurs in the procedure of storing the operation data buffered in the memory into the compound document of the operation object, the “Nothing” effect of the “All of Nothing” transaction features is implemented. Therefore, the highly reliable data storage system is provided for the users.

25 **Brief Description of the Drawings**

Figure 1 is a flowchart illustrating a method for storing data according to a first embodiment of the present invention.

Figure 2 is a flowchart illustrating a method for storing data according to a second embodiment of the present invention.

Figure 3 is a flowchart illustrating a method for storing data according to a third embodiment of the present invention.

Figure 4 is a schematic diagram illustrating a basic structure of an apparatus for storing data according to an embodiment of the present invention.

5 Figure 5 is a schematic diagram illustrating a detail structure of an apparatus for storing data according to an embodiment of the present invention.

Detailed Description of the Invention

In order to make the object, technical solution and merits of the present invention clearer, the present invention will be illustrated in detail hereinafter with reference to the
10 accompanying drawings and specific examples.

A first embodiment is provided.

In order to provide highly reliable data storage, a method for storing data is provided by embodiments of the present invention, as shown in Figure 1, the method includes the following steps.

15 Step 101: operation data of an operation object in a transaction period is received, and the operation data is buffered in a memory.

In the embodiment, the transaction period refers to a period of time needed for implementing a specific transaction. For example, the specific transaction may be a writing operation, several steps are usually needed when implementing the writing
20 operation by the computer, and the steps may include changing the length of the compound document, writing the data into the compound document and so on. Hence, in the embodiment, the period of time needed for implementing the several steps may be configured as a transaction period of the writing operation.

In practical application, the operation object in step 101 may be texts, graphs,
25 spreadsheets, voice and video images etc. in the file system, and the operation performed for the operation object includes but is not limited to a writing operation and a reading operation.

It should be noted that, in step 101, the received operation data of the operation object in the file system may be directly transmitted to a memory of the computer by invoking instructions or pointers of the computer.

Step 102: a backup of original data of the operation object in the compound document is stored in a log file.

When receiving the operation data of the operation object in the file system, the computer determines the original data of the operation object in the compound document, reads the original data from the compound document, and a backup of the original data is stored in the log file. The log file and the compound document are independent, that is, the log file and compound document are stored in different hard disk sectors, or the operations of the log file and the compound document do not affect each other and so on. The log file may be established temporarily, or has been established already, and may be located at the same catalog with the compound document; the log file can store the original data of the operation object before current operation is performed. In the embodiment, the log file is a pronoun of an original data backup file, which may be another name.

Step 103: the operation data buffered in the memory is stored into the compound document.

After the backup of the original data is stored in the log file, the operation of storing the operation data buffered in the memory into the compound document of the operation object may be triggered through sending a backup finish signal or other modes. If the storing in the step 103 is finished successfully, an “All” effect in an “All or Nothing” transaction character (i.e. an operation is performed successfully or no trace is left as nothing happens) is obtained. If the compound document becomes abnormal because of abnormality (such as power-off, dead halt, process crash, deadlock and so on) of applications when step 103 is performed, the original data of the operation object in the compound document is recovered by using the log file in step 102, so that an “Nothing” effect in the “All of Nothing” transaction character is obtained.

As can be seen, the embodiment of the present invention provides a method for storing data, and in the method, the operation data is buffered in the memory firstly;

before the operate data is stored into the compound document, the backup of the original data of the operation object in the compound document is stored in the log file; after the backup is finished, the operation data is stored into the compound document, thereby providing the highly reliable data storage system for the users.

5 The above provides simple descriptions of the method provided by the embodiments of the present invention, and detail descriptions of the method provided by the embodiments of the present invention will be provided through the following two embodiments.

A second embodiment is provided.

10 In order to provide the highly reliable data storage for the users, and to solve the conventional problems, e.g. the damage of the compound document in the hard disk caused by the abnormality of applications in the procedure of storing the data by the file system, embodiments of the present invention provide a method for storing data, and the “All of Nothing” transaction feature of the data operation in the file system is
15 implemented. In the embodiment, the file system processes various operation commands, and the file system may be a file system of a computer in common use, as shown in Figure 2, the method includes the following steps.

Step 201: the computer receives operation data of an operation object in a transaction period, and sends an operation request to the file system.

20 For example, the operation data received by the computer indicates that a writing operation needs to be performed for the operation object, and the computer needs to send a writing operation request to the file system.

Step 202: the file system responses the operation request and stores the received operation data in a memory of the computer;

25 In the prior art, the operation data is directly written into the compound document of the operation object, but in the embodiment, the received operation data is buffered in the memory of the computer firstly, and further indication of the file system is waited for, such as the instruction in following step 209, and the operation data will not be stored into the compound document of the operation object before the further indication of the file

system is received.

Step 203: when the file system responses the operation request, a background thread of the computer requests and obtains an operation monopoly of a log file from an operation system.

- 5 The background thread may be a thread configured in advance and is used to operate the log file.

10 The operation monopoly of the log file may be implemented through mutex variable, the main use of the mutex variable is to avoid the background thread and the upper layer application operate the log file at the same time, so as to ensure that the log file always includes one or multiple integral operations. The mutex variable may be configured in advance by an operator of the computer or may be a mutex variable already exists in the computer in the case of not affecting the normal work of the computer, which is not limited in the embodiments of the present invention.

15 The above mentioned one or multiple integral operations may include the following: when only one operation is needed to implement the backup of the original data of the operation object, the log file only needs to receive one operation to implement the backup of the original data; when multiple operations are needed to implement the backup of the original data of the operation object, the log file needs to receive multiple operations to implement the backup of the original data. For example, if the original data includes only
20 texts, only one operation is needed, e.g. copying the contents of the texts and pasting to the log file; if the original data includes not only texts, but also large numbers of links, pictures, and video etc., multiple operations are needed to implement the backup of the original data. It can be seen that, the use of mutex variable ensures the integrality of the operations for the log file.

25 It should be noted that, besides the mutex variable, other modes may be used to lock the operation monopoly of the log file, which is not limited in embodiments of the present invention. The modes used by those skilled in the art for locking the operation monopoly without creative labor are all belong to the protect scope of the present invention.

Step 204: it is determined whether the log file exists;

If the log file exists, step 205 is performed;

If the log file does not exist, step 206 is performed.

5 The log file may be located at the same catalog with the compound document and used to store the original data of the operation object before the current operation is performed. For example, when the operation object is word, the existing log file may be a file similar to word, the suffix of which is .bak, and when a writing operation is performed for the word, the existing log file is used to store the original data existing before the writing operation.

10 Step 205: a selecting signal sent by a user is received, and whether to cover the existing log file is determined according to the selecting signal;

If the selecting signal indicates not to cover the existing log file, step 206 is performed;

If the selecting signal indicates to cover the existing log file, step 207 is performed;

15 When the determining result in step 204 indicates that the log file exists, a selecting window may be provided to the user in step 205 through a dialog box, tips and other modes, and whether to cover the existing log file may be determined according to the selecting signal sent by the user through the selecting window.

Step 206: a new log file is established; and step 207 is performed.

20 Step 207: the background thread checks the hard disk sectors, obtains the hard disk sector corresponding to the operation object needed to be rewritten in the current transaction period;

In one transaction period, the operation object needed to be rewritten by the computer may be stored in one or more hard disk sectors.

25 Step 208: original data of the operation object needed to be rewritten in the current transaction period is read from the hard disk sector and stored in the log file.

In step 208, storing the original data in the log file may include the following: If the

selecting signal indicates to cover the existing log file in step 205, the original data read from the hard disk sector covers the data of the existing log file; in other situations, the original data read from the hard disk sector is directly written into the log file newly established in step 206.

5 The log file may be opened by using a no-buffer-model; or after the data is written, a function for writing a file into a hard disk sector (such as FlushFileBuffers function) is invoked, so as to ensure that the data can be written into the hard disk sector of the computer integrally. In embodiments of the present invention, the log file may also be opened by using a buffer-model. The difference between the no-buffer-model and the
10 buffer-model includes that: when the file is opened by using the buffer-model, a step of buffering the file into the memory will be included in the process of operating the file, and thus it is not assured that the file is stored into the hard disk sector of the computer; when the file is opened by using the no-buffer-model, the step of buffering the file into the memory is not included, and the file is directly written into the hard disk sector of the
15 computer, so that it is assured that all contents of the file are stored into the hard disk sector of the computer, i.e. the reliability of the log file is ensured.

Optionally, because there are various operations performed for the operation object by the user, the file system may determine whether it is needed to store the backup of the original data by using the log file based on different operations.

20 It should be noted that, if the operation performed for the operation object is a creating operation (i.e. an operation to create a new operation object), the operation object does not have the original data, and it is unnecessary to store the backup of the original data by using the log file, that is, the step 208 is not needed; except the creating operation, if the operation object has the original data, it is needed to store the backup of the original
25 data by using the log file.

Step 209: after the log file is stored, a storing finish instruction is sent to the file system; the file system sends an allow-to-write instruction to the memory, and the operation data buffered in the memory is stored into the compound document of the operation object.

30 In step 209, the file system may store the operation data buffered in the memory into

the corresponding hard disk sector of the computer according to the catalog of the compound document of the operation object. This storing process is a rewriting process of the compound document in a real sense.

Step 210: the log file storing the original data of the operation object is configured as
5 invalid;

If no abnormal situation occurs in the procedure of storing the data, the operation data is stored into the compound document correctly and completely, the log file storing the original data of the operation object is useless, and the log file may be configured as invalid by, such as cancelling the log file. By using the step 210, the usage rate of the
10 storage space of the computer is reduced and wasting of the storage space is avoided.

It should be noted that, the step 210 is an optional step. The step 210 may not be performed, and the step 211 is performed directly after the step 209 is performed. Referring to the log file storing the original data of the operation object, the background thread may clean up the invalid log file when the whole transaction period is finished
15 normally, or the invalid log file is clean up manually or periodically by the operation system.

Step 211: the background thread ends normally, and the obtained operation monopoly of the log file is released.

It should be noted that, if abnormality of applications occurs in the procedure of
20 writing the operation data buffered in the memory into the compound document, i.e. the background thread can not finished normally, specifically if the procedure of storing data is interrupted because of a problem occurs when step 209 is performed, the computer performs the procedure of recovering data, i.e. the file system sends an instruction of recovering data to the log file, and step 209' is performed.

25 Step 209': data of the operation object in the file system is recovered by using the log file storing the original data of the operation object.

It should be noted that, in the procedure of recovering the data, the log file may be verified to ensure the reliability of the log file. In the embodiment, the verification may be Cyclical Redundancy Check (CRC) and so on, which are not described herein. If the

verification is passed, the log file is reliable, and the operation object in the compound document is recovered by using the reliable log file; if the verification is not passed, the log file is unreliable, and the operation object in the compound document is not recovered by using the unreliable log file.

5 As can be seen, in the method for storing data provided by the embodiment of the present invention, if the abnormal situation occurs in the procedure of storing the operation data into the compound document of the operation object, the operation object is recovered by using the original data stored in the log file, so as to provide transactional support for the file system, solve the problem of compound document damage caused by
10 the abnormal situation in the current file system, and provide the highly reliable data storage system; at the same time, the transactional support is provided for upper layer applications, and highly data consistency is maintained; by using the method, each module of the applications does not need to worry about the user data loss or disaccord, and the reliability and integrity of the data of the applications are greatly improved.

15 A third embodiment is provided.

 In order to provide the highly reliable data storage for the users, and to solve the conventional problems, e.g. the damage of the compound document in the hard disk caused by the abnormality of applications, the embodiments of the present invention provide a method for storing data, and an “All of Nothing” transaction feature of the data
20 operation in the file system is implemented. Different from the embodiment shown in Figure 2, in this embodiment, the log file has a flag bit, and the flag bit is used to indicate whether the log file is valid, and in the embodiment, instructions cooperate with each other to store the data, and the instructions may be instructions of a computer in common user. As shown in Figure 3, the method includes the following steps.

25 Step 301: the computer receives operation data of an operation object in a transaction period, if the operation data relates to a file system, the received operation data is directly buffered in a memory of the computer.

 Step 302: a background thread of the computer requests and obtains an operation monopoly of a log file from an operation system.

Step 303: it is determined whether a log file exists;

If the log file exists, step 304 is performed;

If the log file does not exist, step 305 is performed.

Step 304: it is determined whether the flag bit of the log file is invalid;

5 If the flag bit of the log file is invalid, step 306 is performed;

If the flag bit of the log file is not invalid, it is indicated that an abnormal situation occurs when the operation for the compound document is performed last time, and a recovery is needed, the procedure of the method is terminated, or step 309' is performed directly.

10 The flag bit of the log file is used to indicate whether the log file is valid. If the procedure of storing data performed by the computer is successful, and no abnormal situation, such as power-off, dead halt, process crash, deadlock occurs, the original data of the operation object stored in the log file before the current operation is performed is not useful, and thus the flag bit of the log file is configured as invalid; if an abnormal
15 situation occurs in the procedure of storing data performed by the computer, the operation object may be recovered according to the original data of the operation object stored in the log file before the current operation is performed, thereby providing reliable data for subsequent operations.

Step 305: the log file is created, and the flag bit of the log file is configured as
20 invalid, and step 306 is performed.

Step 306: the background thread checks hard disk sectors of the computer, obtains the hard disk sector corresponding to the operation object needed to be rewritten in current transaction period;

Step 307: the original data of the operation object needed to be rewritten in current
25 transaction period is read from the hard disk sector and stored in the log file, and the flag bit of the log file is configured as valid;

Step 308: it is determined whether the log file is stored successfully;

If the log file is stored successfully, step 309 is performed;

If the log file is not stored successfully, step 307 is performed.

5 If the log file is stored successfully, an operation authority of the compound document is open, and the computer can store the data buffered in the memory into the compound document; if the log file is stored unsuccessfully, the operation authority of the compound document is closed, the computer can not operate the compound document, that is, can not store the data buffered in the memory into the compound document until the log file is stored successfully

10 Step 309: the operation data buffered in the memory is stored into the compound document of the operation object in the hard disk sector, and the flag bit of the log file is configured as invalid;

Step 310: the background thread ends normally, and the obtained operation monopoly of the log file is released.

15 It should be noted that, if an application abnormality occurs in the procedure of storing the operation data buffered in the memory into the compound document, i.e. the procedure of storing the data is interrupted when the step 309 is performed, the computer begins to recover the data, i.e. performs the step 309'.

Step 309': the operation object in the compound document is recovered by using the log file with valid flag bit.

20 It should be noted that, the methods for storing data provided by the embodiments of the present invention are not limited by the embodiment shown in Figure 2 in which the file system processes the operation commands to store the data, and are not limited by the embodiment shown in Figure 3 in which the instructions of the computer cooperate with each other to store the data, and any instruction processes that can implement the method
25 for storing the data is within the protection scope of the present invention.

As can be seen, in the method for storing data provided by the embodiment of the present invention, the operation data is firstly buffered in the memory, and the backup of the original data is stored before the operation data is written into the compound

document. In the procedure of storing the data, if the abnormal situation occurs in the procedure of storing the operation data into the compound document of the operation object, the operation object is recovered by using the original data stored in the log file, so as to provide the transactional support for the file system, solve the problem of the compound document damage caused by the abnormal situation in the current file system, and provide the highly reliable data storage system for the users; at the same time, the transactional support is provided for upper layer applications, and highly data consistency is maintained; by using the method, each module of the applications does not need to worry about the user data loss or disaccord, and the reliability and integrality of the data of the applications are greatly improved.

The methods provided by the embodiments of the present invention are described as above, and an apparatus provided by the embodiments of the present invention are described hereinafter.

Figure 4 is a schematic diagram illustrating a basic structure of an apparatus for storing data according to an embodiment of the present invention. The working procedure of the apparatus may refer to the methods described in the above three embodiments. As shown in Figure 4, the apparatus includes:

a receiving module 401, adapted to receive operation data of an operation object in a transaction period, and buffer the operation data in a memory;

a backup module 402, adapted to store a backup of original data of the operation object in a compound document in a log file;

a storing module 403, adapted to store the operation data buffered in the memory into the compound document after the backup module 402 stores the backup of the original data of the operation object in the compound document in the log file.

Figure 4 shows a basic structure of the apparatus provided by the embodiments, and detailed descriptions are provided hereinafter.

Figure 5 is a schematic diagram illustrating a detail structure of an apparatus for storing data according to an embodiment of the present invention. As shown in Figure 5, the apparatus includes: a receiving module 501, a backup module 502 and a storing

module 503. The functions of the receiving module 501, the backup module 502 and the storing module 503 are similar with that of the receiving module 401, the backup module 402 and the storing module 403, and will not be describe herein.

In an embodiment, the backup module 502 includes:

5 a determining unit 5021, adapted to determining whether the operation data received by the receiving module 401 is data of a creating operation;

 a backup unit 5022, adapted to store the backup of the original data of the operation object in the compound document in the log file when the determining unit 5021 determines the operation data is not the data of the creating operation.

10 The backup unit 5022 may store the backup through the following two modes, which are merely examples to make the present invention clearer, and not used to limit the present invention.

 In one mode:

 the backup unit 5022 is adapted to obtain the original data of the operation object in
15 the compound document, open the log file by using a no-buffer-model, and store the backup of the original data of the operation object in the opened log file.

 In the other mode:

 the backup unit 5022 is adapted to obtain the original data of the operation object in the compound document, store the backup of the original data of the operation object in
20 the log file, and store the log file in a hard disk sector by invoking a function.

In another embodiment, the apparatus also includes:

 a operation managing module 504, adapted to request and obtain an operation monopoly of the log file before the backup unit 5022 stores the backup of the original data of the operation object in the log file; and release the operation monopoly of the log
25 file after the backup unit 5022 stores the backup of the original data of the operation object in the log file.

In another embodiment, the apparatus also includes:

a recovering module 505, adapted to recover data of the operation object in the compound document by using the backup of the original data of the operation object in the log file when the compound document becomes abnormal in the procedure of storing the operation data buffered in the memory into the compound document.

Further, the recovering module 505 includes:

a verifying unit 5051, adapted to determine whether the log file is valid; the log file is configured as invalid when the storing module stores the operation data buffered in the memory into the compound document successfully, and the log file is configured as valid when the storing module stores the operation data buffered in the memory into the compound document unsuccessfully;

a recovering unit 5052, adapted to recover data of the operation object in the compound document by using the backup of the original data of the operation object in the log file when the verifying unit 5051 determines that the log file is valid.

It should be noted that, in the apparatus for storing data provided by the embodiments of the present invention, the cooperation may be performed through commands of the computer operation system, or through the file system, which are not used to limit the protection scope of the present invention. Detailed cooperation may refer to the methods for storing data provided by any embodiments shown in Figures 1-3, and are not described herein.

The embodiments of the present invention provide the apparatus for storing data, the operation data is firstly buffered in the memory, and the backup of the original data is stored before the operation data is written into the compound document, and then the operation data buffered in the memory is stored in the compound document of the operation object. In the procedure of storing the data, if the abnormal situation occurs in the procedure of storing the operation data into the compound document of the operation object, the operation object is recovered by using the original data stored in the log file, so as to provide the transactional support for the file system, solve the problem of the compound document damage caused by the abnormal situation in the current file system,

and provide the highly reliable data storage system for the users; at the same time, the transactional support is provided for upper layer applications, and highly data consistency is maintained; by using the method, each module of the applications does not need to worry about the user data loss or disaccord, and the reliability and integrity of the data
5 of the applications are greatly improved.

The foregoing are only preferred embodiments of the present invention and are not for use in limiting the protection scope of the present invention. Any modification, equivalent replacement and improvement made within the scope of the present invention should be covered under the protection scope of the present invention.

Claims

1. A method for storing data, comprising:

receiving operation data of an operation object in a transaction period, and buffering the operation data in a memory;

5 storing a backup of original data of the operation object in a compound document in a log file;

storing the operation data buffered in the memory into the compound document.

2. The method of claim 1, wherein storing a backup of original data of the operation
10 object in a compound document in a log file comprises:

determining whether the operation data is data of a creating operation, storing the backup of original data of the operation object in the compound document in the log file when the operation data is not the data of the creating operation.

15 3. The method of claims 1 or 2, wherein storing a backup of original data of the operation object in a compound document in a log file comprises:

obtaining the original data of the operation object in the compound document, open the log file by using a no-buffer-model, and store the backup of the original data of the operation object in the log file;

20 or

obtaining the original data of the operation object in the compound document, storing the backup of the original data of the operation object in the log file, and storing the log file in a hard disk sector by invoking a function.

25 4. The method of claim 1, further comprising:

requesting and obtaining an operation monopoly of the log file before storing the backup of the original data of the operation object in the log file; and

releasing the operation monopoly of the log file after storing the backup of the original data of the operation object in the log file.

30

5. The method of claim 1, further comprising:

recovering data of the operation object in the compound document by using the backup of the original data of the operation object in the log file when the compound document becomes abnormal in the procedure of storing the operation data buffered in the memory into the compound document.

5

6. The method of claim 5, further comprising:

configuring the log file as invalid when the operation data buffered in the memory is stored into the compound document successfully, and configuring the log file as valid when the operation data buffered in the memory is stored into the compound document
10 unsuccessfully

7. The method of claim 6, wherein recovering data of the operation object in the compound document by using the backup of the original data of the operation object in the log file comprises:

15 determining whether the log file is valid; if the log file is valid, recovering the data of the operation object in the compound document by using the backup of the original data of the operation object in the log file.

8. An apparatus for storing data, comprising:

20 a receiving module, adapted to receive operation data of an operation object in a transaction period, and buffer the operation data in a memory;

a backup module, adapted to store a backup of original data of the operation object in a compound document in a log file;

25 a storing module, adapted to store the operation data buffered in the memory into the compound document after the backup module stores the backup of the original data of the operation object in the compound document in the log file.

9. The apparatus of claim 8, wherein the backup module comprises:

30 a determining unit, adapted to determining whether the operation data is data of a creating operation;

a backup unit, adapted to store the backup of the original data of the operation object in the compound document in the log file when the determining unit determines the operation data is not the data of the creating operation.

10. The apparatus of claim 9, wherein

the backup unit is adapted to obtain the original data of the operation object in the compound document, open the log file by using a no-buffer-model, and store the backup
5 of the original data of the operation object in the log file; or

the backup unit is adapted to obtain the original data of the operation object in the compound document, store the backup of the original data of the operation object in the opened log file, and store the log file in a hard disk sector by invoking a function.

11. The apparatus of claims 9 or 10, further comprising:

a operation managing module, adapted to request and obtain an operation monopoly of the log file before the backup unit stores the backup of the original data of the operation object in the log file; and release the operation monopoly of the log file after the backup unit stores the backup of the original data of the operation object in the log file.

15

12. The apparatus of claim 8, further comprising:

a recovering module, adapted to recover data of the operation object in the compound document by using the backup of the original data of the operation object in the log file when the compound document becomes abnormal in the procedure of storing
20 the operation data buffered in the memory to the compound document.

13. The apparatus of claim 12, wherein the recovering module comprises:

a verifying unit, adapted to determine whether the log file is valid; the log file is configured as invalid when the storing module stores the operation data buffered in the memory into the compound document successfully, and the log file is configured as valid
25 when the storing module stores the operation data buffered in the memory into the compound document unsuccessfully;

a recovering unit, adapted to recover data of the operation object in the compound document by using the backup of the original data of the operation object in the log file
30 when the verifying unit determines that the log file is valid.