(12) UK Patent Application (19) GB (11) 2510178 (13) A

(43) Date of A Publication **30.07.2014**

(21) Application No: **1301498.0**

(22) Date of Filing: **28.01.2013**

(71) Applicant(s):
1&1 Internet AG
(Incorporated in the Federal Republic of Germany)
Elgendorfer Straße 57, Montabaur 56410, Germany

(72) Inventor(s):
Thomas Schöbel-Theurer

(74) Agent and/or Address for Service:
1&1 Internet AG
Sapporobogen 6-8, DE-80637, München, Germany

(51) INT CL:
G06F 17/30 (2006.01)

(56) Documents Cited:
US 6321234 B1 **US 20070185938 A1**
US 20070162516 A1 **US 20040098425 A1**

(58) Field of Search:
INT CL **G06F**
Other: **Online: WPI & EPODOC**

(54) Title of the Invention: **System and method for replicating data**
Abstract Title: **System and method for replicating data**

(57) A system and method for replicating data comprises a first computing device 200 and a second computing device 210 each having non-volatile storage 201, 212. The first computing device receives one or more write requests 214, buffers the write requests in a volatile memory and starts appending 203 the data to a log file 202. After the data is successfully appended to the log file, the first computing device promptly signals completion of the received write request(s) and starts writing the data to the local storage 201. The second computing device detects new data to be written, fetches the delta of the remote log file on the first computing device and appends this data to a local copy of the log file 211 accessible by the second computing device. The second computing devices detects new data, fetches the delta of the local copy of the log file, and writes data to its non volatile storage 212.

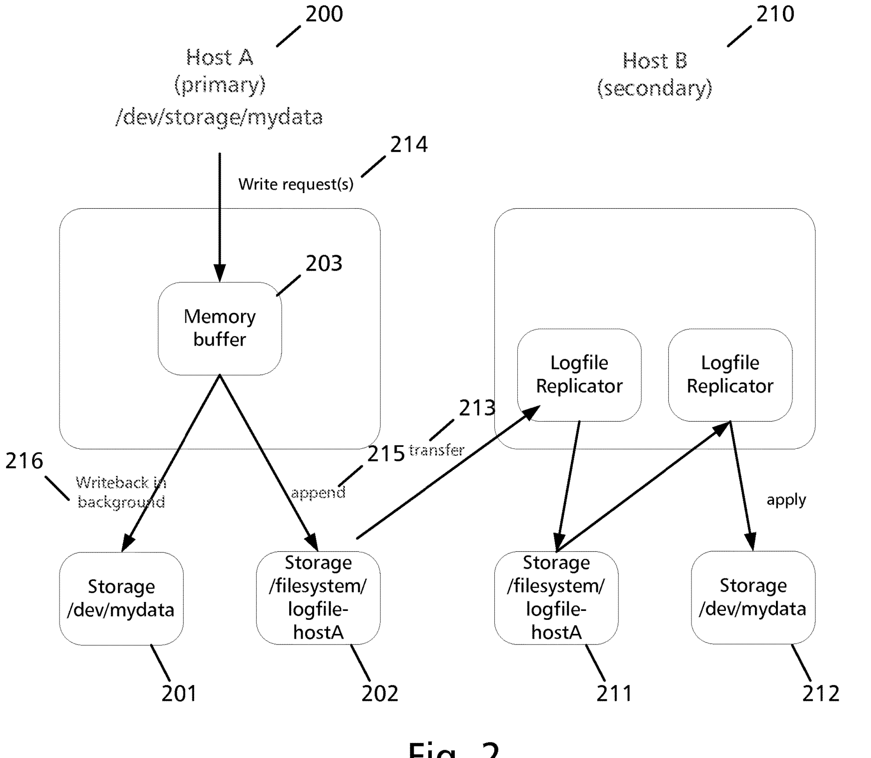

Fig. 2

GB 2510178 A

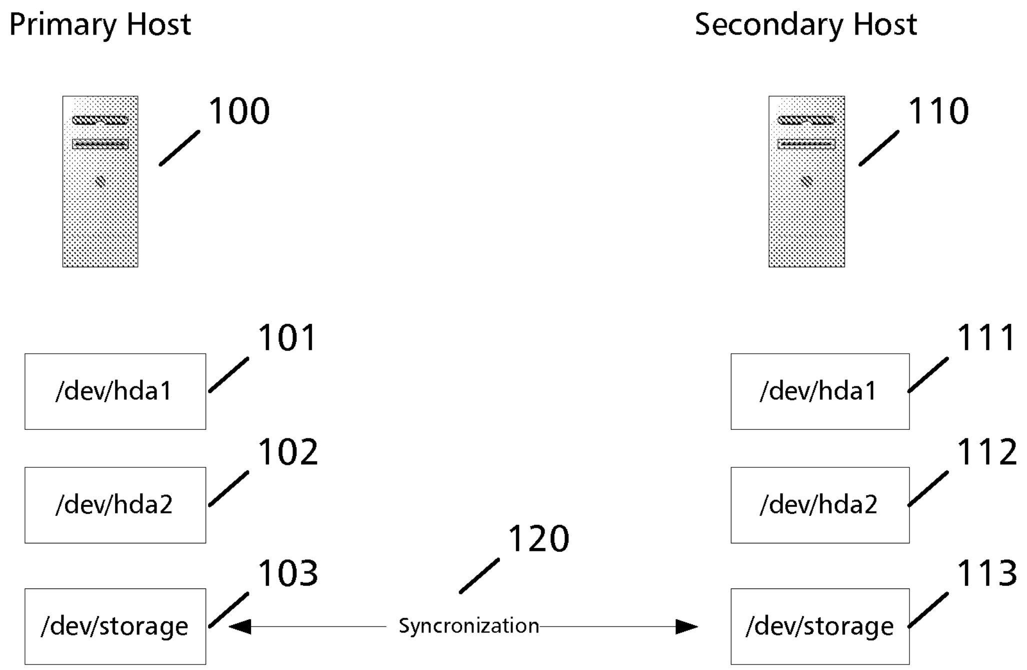

Fig. 1
(prior art)

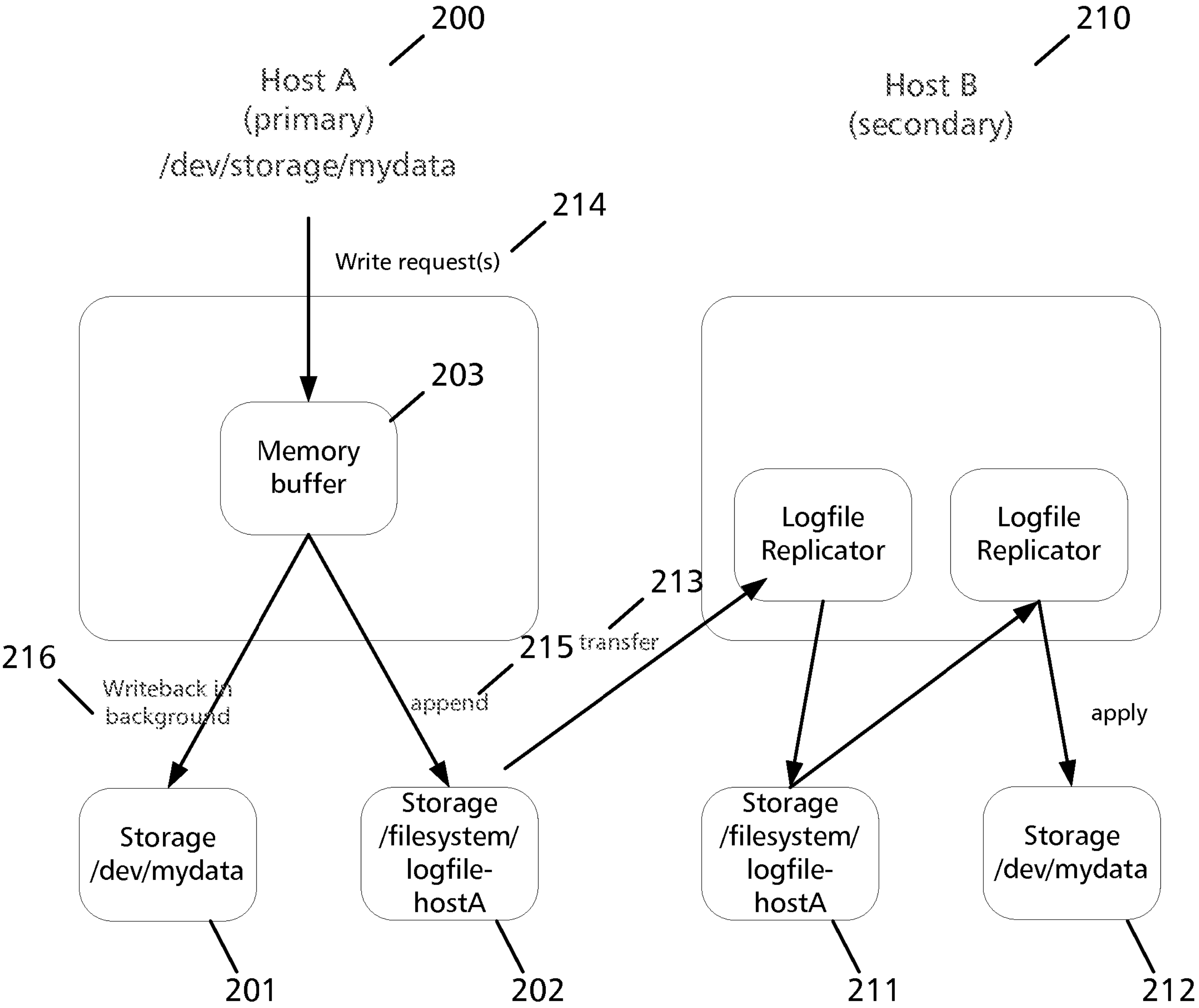
200
Host A
(primary)
/dev/storage/mydata
210
Host B
(secondary)
214
Write request(s)
203
Memory
buffer
Logfile
Replicator
Logfile
Replicator
213
transfer
215
append
216
Writeback in
background
apply
Storage
/dev/mydata
Storage
/filesystem/
logfile-
hostA
Storage
/filesystem/
logfile-
hostA
Storage
/dev/mydata
201
202
211
212

Fig. 2

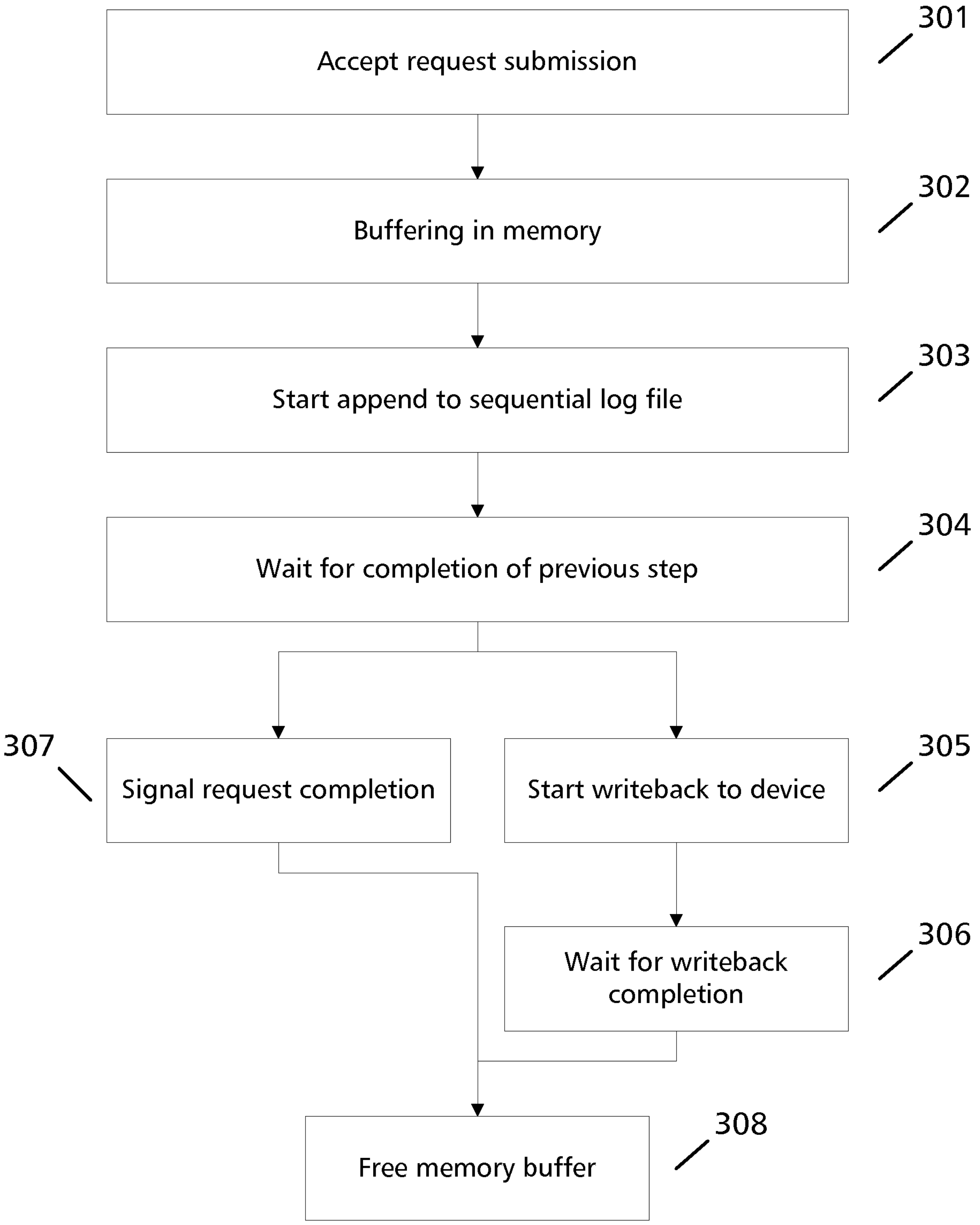
Accept request submission
301
Buffering in memory
302
Start append to sequential log file
303
Wait for completion of previous step
304
307
Signal request completion
Start writeback to device
305
Wait for writeback completion
306
Free memory buffer
308

Fig. 3

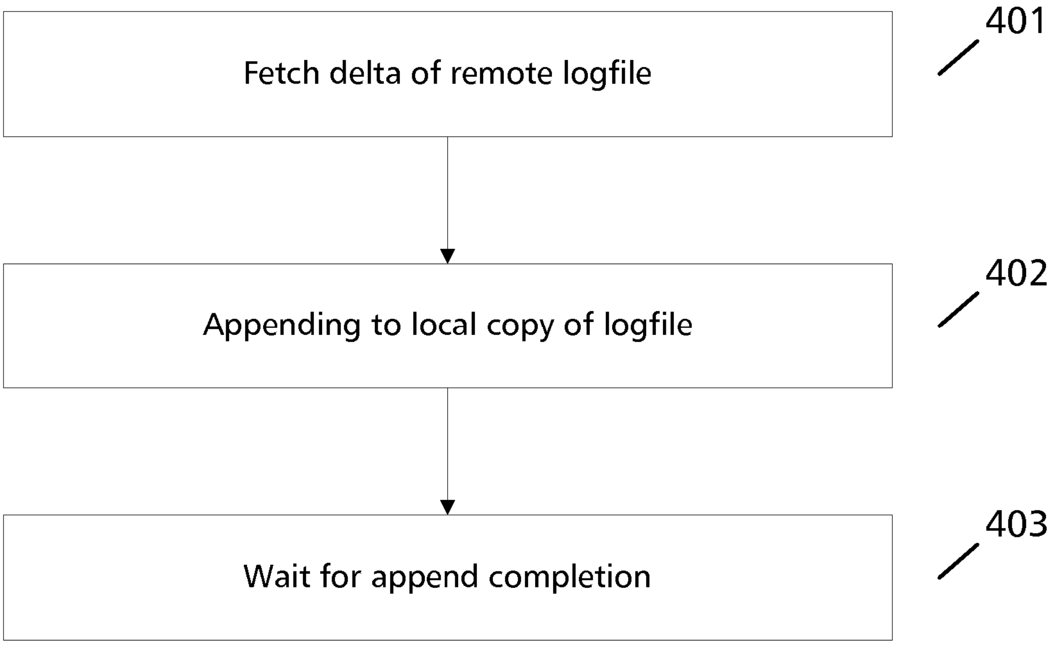
Fetch delta of remote logfile
401
Appending to local copy of logfile
402
Wait for append completion
403

Fig. 4

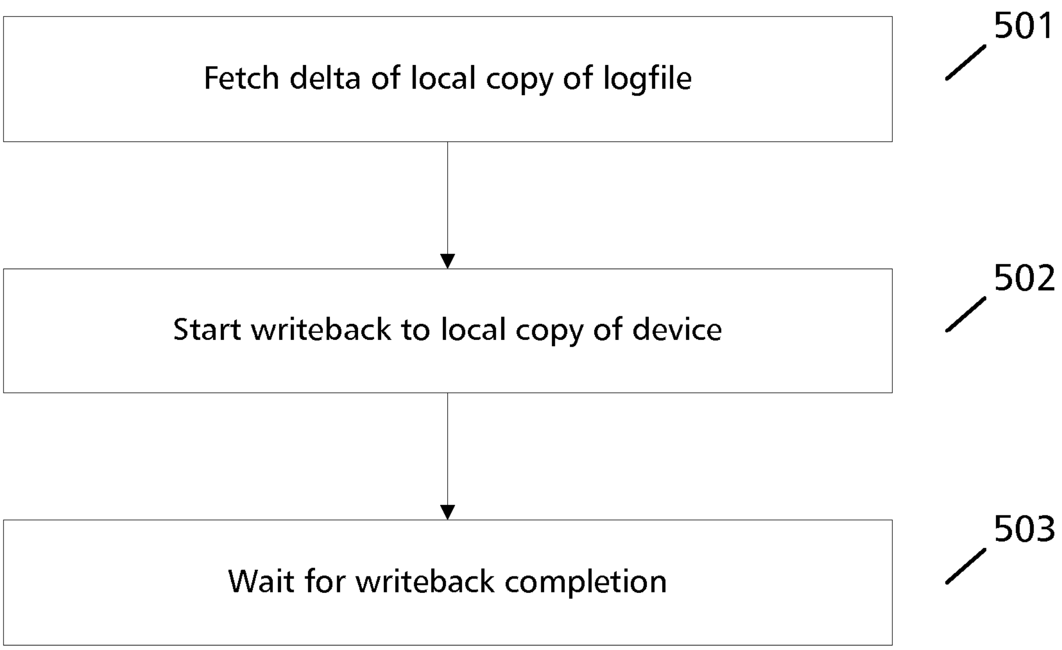
501
Fetch delta of local copy of logfile
502
Start writeback to local copy of device
503
Wait for writeback completion

Fig. 5

System and method for replicating data

The present invention relates to a system and a method for replicating data, in particular to data stored on non-volatile storage media. Every storage media is subjected to eventual temporal or permanent failure – With removable media there is a risk of lost or misplacement, typically stationary media such as hard disks are subject to wear and tear over time. Typically users try to mitigate the risk of losing data by periodically copying data from one storage medium to another. Ideally both media are not subject to the same risk, i.e. they are kept at separate places to ensure that they are not impacted by the same event such as fire, flooding etc. Computer systems and software for computer systems aid the user in keeping backups of his data, e.g. by means of an archiving or backup software that periodically either reminds the user to backup his data on a separate medium performs such a backup automatically based on preset parameters. In case of a failure of the primary media the user either replaces the primary medium with one of the backup media, i.e. the selected backup medium becomes the new primary medium. As an alternative, a new empty primary media is supplied and data is copied or restored from one of the backup media.

In case not only data integrity but also data availability matters, there are techniques known such as RAID systems, that continually mirror data from one storage medium to at least one other, e.g. RAID-1 defines implements such a mirror process which ensures that upon failure of the primary medium data is retrieved from a synchronized copy on a second medium without interruption of the service.

For professional services, high availability clusters are known utilizing for example Distributed Replicated Block Devices (DRBD) for a distributed storage system. DRBD essentially defines RAID-1 type capabilities over a network, thus provides for spatial redundant storage. In more detail, DRBD provides for logical block devices acting as a proxy for the physical local block devices in each storage node. Data written to the logical block device on the primary node is transferred to a physical block device in the primary node and subsequently copied to the secondary node. In case of a storage media failure at the primary node, the secondary node is promoted to become the new primary node. However, this transition results in inconsistencies between the data stored in the primary and secondary not and thus requires a subsequent verification of the integrity of the file system stacked on top of DRBD or the deployment of a journaling file system such as EXT4.

Generally, systems deploying DRBD require a network connection between each storage systems hat exhibits high and constant data throughput typically only found in managed local area networks. DRDB fails to perform well in wide area network with varying data throughput.

It is the object of the present invention to provide a system and a method for replicating data that is tolerant to unreliable network links and that provides anytime data consistency.

This object is solved by the subject matter of the independent claims. Preferred embodiments are defined by the dependent claims.

The present invention is related to a system for replicating data comprising a first computing device having a first storage means, a second storage means and volatile memory. The inventive system further comprises a second computing device having a third and a forth storage means, said third storage means being communicatively coupled to the second storage means. In a preferred embodiment of the invention the first computing device has means for receiving write requests, each request containing payload data, means for writing said payload data of each write request to the volatile memory and appending said each payload data to the second storage means, means for acknowledging said write request prior to writing the payload data to the second storage means. Moreover, the second computing device has means for detecting new data in the third storage means and means for applying said new data to the forth storage means.

According to an embodiment, appending each payload data to the second storage means includes appending each payload data to the second storage means in a different order than it was received.

According to an embodiment, appending each payload data to the second storage means includes appending each payload data to the second storage means in ascending block number order.

Brief description of the figures:

Figure 1 illustrates a system for replicating data according to prior art. Prior Art systems comprise a primary host (100) having multiple storage means (101, 102) such as block level devices that are exposed as a device file (103) that is exposed for read and write access. In order to provide data redundancy, a secondary host (110) also having multiple storage means (111, 112) mapped to a device file (113) is synchronized (120) with the device file (103) of the primary host.

Figure 2 illustrates a block diagram of an exemplary embodiment of a system for replicating data according to the present invention.

Figure 3 illustrates a flow chart of an exemplary embodiment for processing data on the primary computing device.

Figures 4 and 5 illustrate flow charts of an exemplary embodiment for processing data on the secondary computing device.

Detailed description of the invention:

According to an embodiment of the invention a first computing device (200) and a second computing device (210) are communicatively coupled, each computing device having local non physical non-volatile storage (201, 212) that is suited for storing block level data. The physical non-volatile storage, such a single hard disk or a bunch of hard disk logically linked with each other is exposed to the outside using a pseudo device. The pseudo devices refer to an arbitrary device node, such as /dev/mydata as shown in Figure 2.

The first computing device receives one or more write requests (214, 301), each write request containing meta data such as an op code, a timestamp, a length indication and payload data – said may for example correspond to a full or a portion of a data block to be stored on a block level device.

As shown in Fig.3 the first computing device upon receiving (301) one or more write requests buffers (302) the write requests in a volatile memory such as a RAM, and starts appending (303) the data to a sequentially organized log file accessible from or attached to the first computing device. After the data was successfully appended to the log file, the first computing device promptly signals (307) completion of the received write request(s) and starts writing (305) the data to the local storage (201). Once this write operation has successfully completed (306), the buffer space used in the volatile memory is freed.

According to an embodiment, the first computing device may append the data of the write requests to the log file in different order than it was originally received. In particular, it may order the request in ascending or descending order with respect to the block numbers obtained from the metadata or payload data in each write request. Also, if a write request refers to a block number that has been received previously but not yes completed appending to the log file, the first computing device may process only the last write request for a particular block number and disregard the previous one, for example by freeing the buffer for said previous write request.

According to an embodiment illustrated by Fig. 4 the second computing device detects new data to be written, fetches (401) the delta of the remote log file on the first computing device and appends (402) this data to a local copy of the log file accessible by the second computing device.

It should be noted that the step of detecting new data can be performed by several variants:
According to an aspect of the invention, the second computing devices open a connection oriented network connection (such as a TCP connection) to the first computing devices over which either the second computing device pulls for new data from the first computing device or the first computing devices pushes new data to the second computing device through this connection. According to a different aspect of the invention the two log files (202, 211) may be provided by a shared file system, allowing both the first and the second computing device to access the same file - in this case, appropriate file pointers must be used to determine the respective read/write positions within the file. According to yet a different aspect of the invention, the two log files may be synced by an external syncing tool, such as rsync known in the Linux computing operating system.

According to an embodiment illustrated by Fig. 5 the second computing devices detects new data to be written, fetches (501) the delta of the local copy of the log file, starts write back (502) to a local non volatile storage.

It should be noted, that each process defined by the process steps in each of Figure 3, 4 and 5 may be performed in parallel, for example the first computing device may continue writing the payload of new write requests to the local log file while the second computing device is still busy accessing log file information for the previous requests.

According to an embodiment there may be more than on secondary computing devices, each having local storage and the ability to perform the steps previously disclosed for the second computing device.

The previous disclose assumes normal operating conditions, in particular at least a working network connection between the first and second computing device and sufficient memory for storing all necessary data. The inventive concept shall, however, also cover the following particular cases:

If there is no more space left for writing write requests to the log file (step 303 fails) the first computing devices attempts to free memory by rotating the log file, i.e. moving the log file to a storage with sufficient free space. If this attempt is unsuccessful or provides only temporary relief, the first computing devices switches to a “by-pass mode” in which the first computing device refrains from writing to log files and starts writing data directly to the local storage.

If the transfer of the log file from the first computing device to the second computing device results in an incomplete log file at the second computing device, i.e. caused by a faulty network connection, the secondary computing device will attempt a fast full sync based on the last known to be good data in the local log file.

In case of a so-called split-brain condition, i.e. when the checksums of the log files do not match because of a faulty operation at the first computing device, the system requires manual intervention by setting the appropriate computing device to become the new primary computing device, followed by a full sync of the newly appointed secondary computing devices.

***

Claims

1. A system for replicating data comprising:
   a first computing device (200) having a first storage means (201), a second storage means (202) and volatile memory (203), the system further comprising
   a second computing device having a third storage means (211) and a forth storage means (212), said third storage means being communicatively coupled (213) to the second storage means,
   wherein the first computing device has
   means for receiving write requests (214), each request containing payload data,
   means for writing said payload data of each write request to the volatile memory and appending (215) said each payload data to the second storage means,
   means for acknowledging said write request prior to writing (216) the payload data to the second storage means, and
   wherein the second computing device has
   means for detecting new data in the third storage means and
   means for applying (217) said new data to the forth storage means.

2. The system of claim 1 wherein appending (215) each payload data to the second storage means (202) includes appending each payload data to the second storage means in a different order than it was received by the first computing device (202).

3. The system of one of claims 1 or 2 wherein appending (215) each payload data to the second storage means (202) includes appending each payload data to the second storage means in ascending block number order.

4. The system of one of claims 1 to 3, wherein the second storage means (202) is a log file stored in the file system provided by the first computing device (200) and wherein the third storage means (211) is a log file stored in the file system provided by the second computing device (210).

5. The system of one of claims 1 to 4, wherein the second computing device (210) connects to the first computing device (200) using a dedicated connection oriented network connection.

6. A method for replicating data from a first computing device (200) to a second computing device (210) comprising the steps of:
Receiving a write request (214, 301) with data on the first computing device,
Buffering (302) said received data in volatile memory (203) of the first computing device,
Appending (215, 303) the received data to a log file (202) on the first computing device,
Acknowledging (307) the write request,
Accessing (401) the log file from the second computing device, and
Writing (217, 402) the data obtained from the log file to a nonvolatile storage (212) attached to the second computing device.

7. The method of claim 6 wherein accessing the log file from the second computing device (210) comprises copying at least a part of the log file from the first computing device to the second computing device.

8. The method of claim 6 wherein accessing the log file on the second computing device comprises accessing the log file from a file system shared between the first and second computing device.

9. The method of one of claims 6 to 8 wherein the data is a sequence of Bytes having a predetermined length.

10. The method of one of claims 6 to 9 wherein the log file (202, 211) comprises an opcode, a timestamp, data and an indication of the length of the data.

11. A first and a second storage system adapted to perform the method steps of one of claims 6 to 10.

12. A computer readable medium having stored thereon instructions to enable a processor to perform the method steps of one of claims 6 to 10.

***

Application No:	GB1301498.0	**Examiner:**	Stuart Purdy
Claims searched:	1-5 (in part) and 6-12	**Date of search:**	20 November 2013

Patents Act 1977: Search Report under Section 17

Documents considered to be relevant:

Category	Relevant to claims	Identity of document and passage or figure of particular relevance
X	1-12	US 2007/185938 A1 (PRAHLAD) See whole document and note in particular paragraphs 22-25, 105-114, 120-122 and 124 and figure 4;
X	1, 3-12	US 2004/098425 A1 (WISS) See whole document in combination with the teaching of US 6321234 and note in particular paragraphs 18-20 and 61-67;
X	1, 3-12	US 6321234 B1 (DEBRUNNER) See whole document and note in particular column 3 lines 14-48, to be read in combination with US 2004098425;
A	-	US 2007/162516 A1 (THIEL) See whole document and note in particular paragraphs 1, and 16-21

Categories:

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC^X :

Worldwide search of patent documents classified in the following areas of the IPC

G06F

The following online and other databases have been used in the preparation of this search report

WPI & EPODOC

International Classification:

Subclass	Subgroup	Valid From
G06F	0017/30	01/01/2006