Appeal Decision

Appeal No. 2020-2298

Appellant	Datrium Inc.
Patent Attorney	SUGIMURA, Kenji
Patent Attorney	YOSHIZAWA, Yuro

The case of appeal against the examiner's decision of refusal of Japanese Patent Application No. 2015-130381, entitled "DATA STORAGE SYSTEM USING DECENTRALIZED VIRTUAL ARRAY" [the application published on December 15, 2016, Japanese Unexamined Patent Application Publication No. 2016-212819] has resulted in the following appeal decision.

Conclusion

The appeal of the case was groundless.

Reason

No. 1 History of the procedures

The present application was filed on June 29, 2015 (priority claim under the Paris Convention: May 12, 2015 (hereinafter referred to as "Priority date")). The history of the procedures is as follows.

June 20, 2018	: Written request for examination
Dated April 9, 2019	: Notice of reasons for refusal
June 7, 2019	: Submission of Written opinion and Written amendment
Dated November 20, 2019	: Examiner's decision of refusal
February 19, 2020	: Submission of Written appeal and Written amendment
June 23, 2020	: Reconsideration report
September 9, 2020	: Written statement

No. 2 Decision to dismiss amendment on the written amendment submitted on February 19, 2020

[Conclusion of Decision to Dismiss Amendment]

The written amendment submitted on February 19, 2020 (hereinafter referred to as "the Amendment") shall be dismissed.

[Reason]

1 Regarding the Amendment (Details of Amendment)

(1) Description of the scope of claims after the amendment

The Amendment amended the recitation of Claim 1 of the scope of claims as follows. (Amended portions are underlined. The claim recited in the scope of claims is hereinafter referred to as "the Claim after the amendment".)

"[Claim 1]

A data storage system comprising:

a plurality of host servers (100), each of which includes at least one processor (111) and which communicates with a plurality of storage nodes (310) over at least one network (200),

at least one storage node including at least one persistent storage device (340-1, 340-N) and at least one storage controller (320-1, 320-2, ..., 320-N),

the host servers including:

at least one process that issues data storage read/write requests;

a memory device forming a cache (113) configured for caching data corresponding to a write request in a non-persistent manner,

a system software (115) component configured to write data corresponding to the write request at least twice; and

a distributed virtual array (DVA) sub-system (405) comprising computer-executable code which, when executed on the at least one processor, causes at least one processor to execute at least one storage processing function on the data corresponding to the write request, wherein

the system software (115) component,

in first write processing, writes data corresponding to the write requests from the host servers to at least one remote non-volatile memory device (330) of the storage nodes (310) without involving any of the persistent storage devices (340-1, 340-N) of the storage nodes (310),

in second write processing, writes data generated by the at least one storage processing function on the data corresponding to the write request to the at least one persistent storage device (340-1, 340-N) of the storage nodes (310) without involving the at least one remote non-volatile memory device (330),

the at least one remote non-volatile memory device (330) constitutes a temporary backup data storage system,

the at least one storage processing function includes at least one of:

computing a fingerprint of data written and storing the fingerprinted data in a selected one of the storage nodes (310);

compressing data to be written and sending the data in compressed form for storage in at least one of the storage nodes (310);

computing error-correction data over the data to be written and storing both the data to be written and the computed error-correction in at least one of the storage nodes (310); and

logging data writes and transmitting corresponding logging information to at least one node's storage controller (320-1, 320-2, ..., 320-N)."

(2) The scope of claims before the Amendment

The recitation of Claim 1 of the scope of claim before the Amendment, the claim having been amended by the written amendment submitted on June 7, 2019, is as follows. (The claim recited in the scope of claims is hereinafter referred to as "the Claim before the amendment").

"[Claim 1]

A data storage system comprising:

a plurality of host servers (100), each of which includes at least one processor (111) and which communicates with a plurality of storage nodes (310) over at least one network (200),

at least one storage node including at least one persistent storage device (340-1, 340-N) and at least one storage controller (320-1, 320-2, ..., 320-N),

the host servers including:

at least one process that issues data storage read/write requests;

a memory device forming a cache (113) configured for caching data corresponding to a write request in a non-persistent manner,

a system software (115) component configured to write data corresponding to the write request to at least one remote non-volatile memory device (330) as a temporary storage device; and

a distributed virtual array (DVA) sub-system (405) comprising computer-executable code which, when executed on the at least one processor, causes at least one processor to execute at least one storage processing function on the data corresponding to the write request, wherein

the at least one storage processing function includes at least one of:

computing a fingerprint of data written and storing the fingerprinted data in a selected one of the storage nodes (310);

compressing data to be written and sending the data in compressed form for storage in at least one of the storage nodes (310);

computing error-correction data over the data to be written and storing both the data to be written and the computed error-correction in at least one of the storage nodes (310); and

logging data writes and transmitting corresponding logging information to at least one node's storage controller (320-1, 320-2, ..., 320-N)."

2 Propriety of amendment

The Amendment adds the following limitations regarding the write processing of the "software (115) component" which is a matter required for specifying the invention recited in Claim 1 before the amendment:

"at least twice" for write processing, and

"in first write processing, writes data corresponding to the write requests from the host servers to at least one remote non-volatile memory device (330) of the storage nodes (310) without involving any of the persistent storage devices (340-1, 340-N) of the storage nodes (310),

in second write processing, writes data generated by the at least one storage processing function on the data corresponding to the write request to the at least one persistent storage devices (340-1, 340-N) of the storage nodes (310) without involving the at least one remote non-volatile memory device (330),

the at least one remote non-volatile memory device (330) constitutes a temporary backup data storage system".

The invention recited in Claim 1 before the amendment and the invention recited in Claim 1 after the amendment belong to the same industrial field and aim to solve the same problems. Thus, the Amendment is to aim at the restriction of the scope of claims stipulated in Article 17-2(5)(ii) of the Patent Act.

We will examine below as to whether the invention recited in Claim 1 after the amendment (hereinafter referred to as "the Amended Invention") falls under the provisions of Article 126(7) of the Patent Act which is applied mutatis mutandis pursuant to Article 17-2(6) of the Patent Act (whether or not the Appellant can be granted a patent independently at the time of filing of the patent application).

(1) the Amended Invention

The Amended Invention recited in Claim 1 in "1 (1)" is as follows (shown again).

"A data storage system comprising:

a plurality of host servers (100), each of which includes at least one processor (111) and which communicates with a plurality of storage nodes (310) over at least one network (200),

at least one storage node including at least one persistent storage device (340-1, 340-N) and at least one storage controller (320-1, 320-2, ... , 320-N),

the host servers including:

at least one process that issues data storage read/write requests;

a memory device forming a cache (113) configured for caching data corresponding to a write request in a non-persistent manner,

a system software (115) component configured to write data corresponding to the write request at least twice; and

a distributed virtual array (DVA) sub-system (405) comprising computer-executable code which, when executed on the at least one processor, causes at least one processor to execute at least one storage processing function on the data corresponding to the write request, wherein

the system software (115) component,

in first write processing, writes data corresponding to the write requests from the host servers to at least one remote non-volatile memory device (330) of the storage nodes (310) without involving any of the persistent storage devices (340-1, 340-N) of the storage nodes (310),

in second write processing, writes data generated by the at least one storage processing function on the data corresponding to the write request to the at least one persistent storage device (340-1, 340-N) of the storage nodes (310) without involving the at least one remote non-volatile memory device (330),

the at least one remote non-volatile memory device (330) constitutes a temporary backup data storage system,

the at least one storage processing function includes at least one of:

computing a fingerprint of data written and storing the fingerprinted data in a selected one of the storage nodes (310);

compressing data to be written and sending the data in compressed form for storage in at least one of the storage nodes (310);

computing error-correction data over the data to be written and storing both the data to be written and the computed error-correction in at least one of the storage nodes (310); and

logging data writes and transmitting corresponding logging information to at least one node's storage controller (320-1, 320-2, ..., 320-N)."

(2) Described matters in the Cited Document

A　Technical matters described in Cited Document 1 and Cited Invention

(A) The specification of U. S. Patent Application Publication No. 2015/0127975 (The application was published on May 7, 2015, the document is hereinafter referred to as "Cited Document 1"), which was cited in the reasons for refusal stated in the examiner's decision and distributed or made publicly available through an electric telecommunication line prior to the priority date of the present application, describes the following technical matters with related drawings.
(Note by the body: The underlines were added by the body for reference.　The same applies hereinafter.)
A "[0014] Embodiments of a distributed virtual array data storage system and method are disclosed.　Storage nodes made up of relatively inexpensive disks with associated processors are scalable to store very large amounts of data.　The storage nodes communicate with servers directly over a network through, for example, an Ethernet connection.　Control of the storage nodes and access to the storage nodes is handled entirely on the server side of the system by distributed virtual array (DVA) software running on the server side and employing a particular protocol over the standard network connection.　The DVA software facilitates all data access functions for applications running on the servers, as well as all data maintenance functions such as

disk reconstruction, garbage collection, etc.

[0015] FIG. 2 is a block diagram of a data storage system 200 according to an embodiment. System 200 includes multiple servers 204A-204G on the server side, and multiple simple storage nodes 202A-202L on the storage side. The actual numbers of servers and storage nodes shown is arbitrary and chosen for example only. There could typically be many more servers and many more storage nodes. In addition, the system 200 is scalable by adding elements such as servers and storage nodes."

B "[0017] FIG. 3 is a block diagram of one server 304A of multiple servers according to an embodiment. The server 304A includes Flash memory device for a server-side cache 312 (also referred to as local cache), and multiple guests VMs 310A-310C and a driver 308. In various embodiments, the server-side cache 312 could alternatively be any appropriate type of memory device for the purpose, such as any solid state memory device, including without limitation Flash, phase-change random access memory (PRAM), etc., and it could be packaged as a Solid-State Disk (SSD), or a circuit card plugged into a bus within the server, or included as a module on the server motherboard. The server 304A also includes a DVA worker VM 311 that runs the DVA software 313. The DVA software 313 handles all storage-related functions for the guest VMs, which are also referred to as application VMs. The DVA software may implement a file system for the storage of vDisks. The file system may be a block-addressed file system, a content-addressed deduplicating file system, a log-structured file system, or any other type of suitable file system. The DVA software handles reads and writes and other functions on the data path and also performs background storage management functions, such as background reconstruction of entire disks, scrubbing, garbage collection, rebalancing, migration, etc. In other embodiments, the server hosts multiple worker VMs, and these worker VMs cooperate to perform the DVA process, including the background storage management functions, for multiple vDisks. At times there may be vDisks that are not in use by any VM host. In an embodiment, such unused vDisks are reconstructed cooperatively by the multiple worker VMs."

C "[0020] FIG. 5 is a block diagram of one storage node 502A of multiple storage nodes according to an embodiment. The storage node 502A includes multiple disks 501A-501F and a standard CPU 505. The multiple disks 501A-501F are examples of physical data storage devices. There can be any number of disks in a storage node 501. Six disks 501 are shown as an example. The disks 501 themselves can be any appropriate type of physical storage device. In an embodiment, the storage node 502A

also includes non-volatile random-access memory (NVRAM) 510. NVRAM devices retain stored data when the power is lost, and have relatively low latency on write operations. According to embodiments of a DVA data layout scheme, each VM host computer server 204 writes to disjoint sets of physical stripes across the disks 501. Also, new Write data are stored to the NVRAM 510 of at least one of the storage nodes 502 so that if there is a power failure, a copy of the new Write data survives. New Write data can also be stored to the NVRAM 510 of at least two storage nodes 502, so that if one node fails there is a redundant copy of the new Write data. According to an embodiment, the DVA method includes storing new Write data for at least one vDisk to at least one storage node so that if the VM host compute server fails, a copy of the newly written data survives on that storage node. New Write data can also be buffered in the VM host until there is enough data to write a full stripe of new data to storage nodes. New Write data can be buffered on the VM host for even longer periods to give time for later overwrites of the same location in a vDisk to obviate the need to write out the overwritten data in a stripe. Embodiments of the DVA method also include a writing VM host computer server computing at least one erasure coded block for a full stripe of data, and writing the at least one erasure coded block to a storage node.
[0021] The CPU 505 performs functions on the disks 501 as directed by the DVA software 513 via the network 206 over a standard network connection 208. For purposes of embodiments described herein, the DVA software 513 could be running anywhere on the server side. However, the DVA software 513 is not envisioned running on the storage side, for example, in the CPU 505, as long as there are servers available to run the software. The relative simplicity of the storage nodes makes the storage nodes inexpensive, eliminates the need for dedicated backplanes and fibre channel connections on the storage side, and allows the entire storage side to communicate with the server side directly via a standard network connection. In various embodiments, the storage nodes are not configured to host VMs, at least in part, because such a capability is never needed. In various embodiments, the storage nodes are not capable of hosting VMs, at least in part, because the inexpensive storage CPUs are not sufficiently powerful to host VMs."

D "[0027] FIGS. 9A and 9B are a flow diagram illustrating a method for a host driver to access data storage according to an embodiment. The driver determines at 902 whether a request for a storage access operation (typically a Read or Write operation) is the first one for this vDisk. If it is the first request, the driver obtains the vDisk map from the vDisk map service (904). Then it is determined (906) whether the operation is a READ or WRITE operation. This sequence is shown as an example. It is also possible to determine the type of operation before it is determined whether the operation is the first storage access request for the application. If it is not the first request, the process determines (906) whether the operation is a READ operation or a WRITE operation.

[0028] If the operation is a WRITE, data are written directly to the NVRAM of one or more of the storage nodes (914). Then new Write data are written to the local cache (916). If a full stripe of Write data is available (930), then an erasure code block is computed for the stripe of new Write data (932), and the Write data plus the erasure code block are written to one or more storage nodes (934). If a full stripe of Write data is not available in 930, new Write data can be buffered in the VM host until there is enough data to write a full stripe of new data to storage nodes. It is also possible to wait longer before writing the stripe to give time for overwrites to obviate the need to write out some data in a stripe. The process then moves to the next instruction at 912."

E "FIG. 3

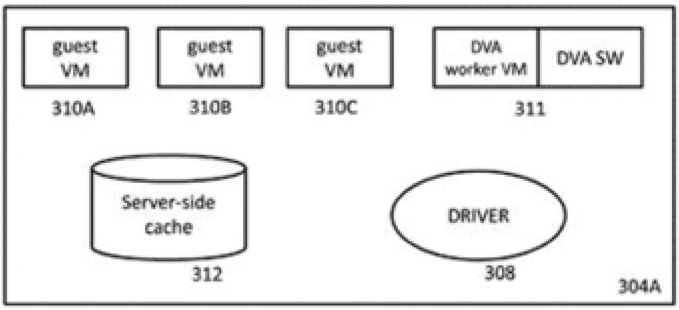

FIG. 3

"

F "FIG. 5

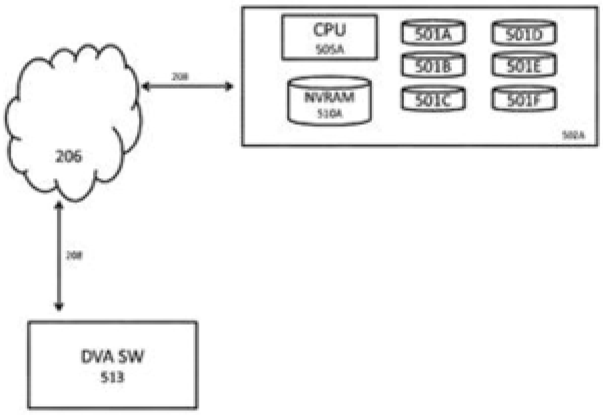

FIG. 5

"

G "FIG. 9A

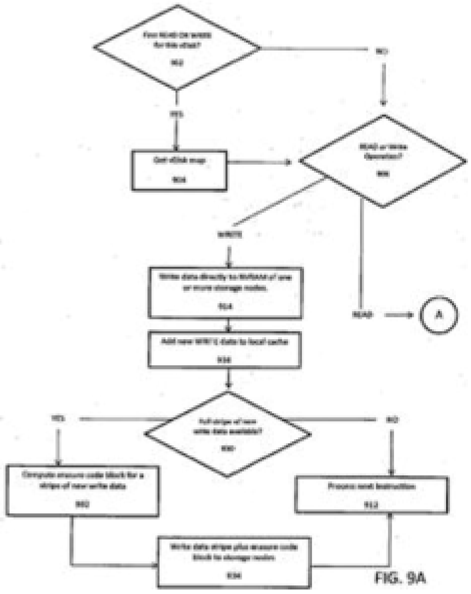
First READ OR WRITE for this vDisk?
902
NO
YES
Get vDisk map
904
READ or Write Operation?
906
WRITE
Write data directly to NVRAM of one or more storage nodes.
914
READ
A
Add new WRITE data to local cache
916
YES
Full stripe of new write data available?
930
NO
Compute erasure code block for a stripe of new write data
932
Process next instruction
912
Write data stripe plus erasure code block to storage nodes
934

FIG. 9A

"

B　According to the above descriptions A to G, it can be said that Cited Document 1 describes the following invention (hereinafter referred to as "Cited Invention").

"A distributed virtual array data storage system, wherein

control of storage nodes and access to the storage nodes are handled entirely on the server side of a system by distributed virtual array (DVA) software running on the server side,

the system includes multiple servers on the server side, and multiple storage nodes on the storage side,

the server includes a server-side cache (also referred to as local cache) and a driver,

the server includes a DVA worker VM that runs the DVA software, the DVA software handles all storage-related functions for guest VMs, which are also referred to as application VMs,

the storage nodes communicate with the servers over a network,

the storage node includes multiple disks and a CPU,

the storage node includes non-volatile random-access memory (NVRAM),

new Write data are stored to the NVRAM of at least one of the storage nodes so that if there is a power failure, a copy of the new Write data survives,

the CPU performs functions on the disks,

a method for a host driver to access data storage is configured so that,

if the storage access operation (read or write operation normally) is a WRITE, data are written directly to the NVRAM of one or more of the storage nodes, then new Write data are written to the local cache, then an erasure code block is computed for the stripe of new Write data, and the Write data plus the erasure code block are written to one or more storage nodes."

(3) Comparison

A The Amended Invention and the Cited Invention are compared below.

(A) The "distributed virtual array data storage system", the "multiple servers", the "multiple storage nodes", and the "network" in the Cited Invention correspond to the "data storage system", the "plurality of host servers (100)", the "plurality of storage nodes (310)", and the "network (200)" in the Amended Invention, respectively.

Each of the "multiple servers" in the Cited Invention obviously includes a processor for performing information processing as a server, and "the storage nodes communicate with the servers over a network". Thus, the each of the multiple servers in the Cited Invention communicates with a plurality of storage nodes over a network.

Therefore, each of the Amended Invention and the Cited Invention is identified as a "data storage system" "comprising a plurality of host servers (100), each of the host servers includes at least one processor (111) and which communicates with a plurality of storage nodes (310) over at least one network (200)".

(B) The "storage node" in the Cited Invention "includes" "multiple disks". A "disk" is to store data persistently. Thus, the "disks" in the Cited Invention correspond to the "persistent storage devices (340-1, 340-N)" in the Amended Invention.

The "storage node" in the Cited Invention "includes a CPU", and "the CPU performs functions on the disks". Thus, the storage node obviously has an information processing function. Since data "write" processing is performed on a storage node, the storage node also has a storage controller function of some kind for controlling the "write" processing, obviously.

Therefore, each of the Amended Invention and the Cited Invention is identified as a "data storage system in which at least one storage node includes at least one persistent storage device (340-1, 340-N) and at least one storage controller (320-1, 320-2, ... , 320-N)".

(C) The Cited Invention is configured so that "the server" "includes a driver" and "a host driver accesses data storage", and the Cited Invention performs "the storage access operation (read or write operation normally)". Thus, the host driver of the server obviously executes at least a process to issue a read or write request to the data storage.

Therefore, the Amended Invention and the Cited Invention are identical in that "a host server " "includes" "at least one process that issues data storage read/write requests".

(D) The Cited Invention is configured so that "the server includes a server-side cache (also referred to as local cache)" and "new Write data are written to the local cache". Thus, the server is considered to include a memory device which caches new write data corresponding to a write request in a non-persistent manner.

Therefore, the Amended Invention and the Cited Invention are identical in that "a host server" "includes" "a memory device forming a cache (113) configured for caching data corresponding to a write request in a non-persistent manner".

(E) In the Cited Invention, the "server" "includes a driver", and when "the host driver accesses data storage", "if the storage access operation (read or write operation normally) is a WRITE, data are written directly to the NVRAM of one or more of the storage nodes", "then" "the Write data plus the erasure code block are written to one or more storage nodes".

The "host driver" of the server is considered as a kind of system software component. The write processing represented by the description, "if ... is a WRITE, data are written directly to the NVRAM of one or more of the storage nodes", is considered as "first write processing". The write processing represented by the description, "then" "the Write data plus the erasure code block are written to one or more storage nodes", is considered as second write processing. Thus, it can be said that the Cited Invention is configured so that a driver, which is a kind of software component included in a server, performs two write processes, such as first and second write processing.

Therefore, the Amended Invention and the Cited Invention are identical in that "a host server" "includes" "a system software (115) component configured to write data corresponding to the write request at least twice", although they have the different features described later.

(F) The Cited Invention is configured so that "control of storage nodes and access to the storage nodes are handled entirely on the server side of a system by distributed virtual array (DVA) software running on the server side", and "the server includes a DVA worker VM that runs the DVA software, the DVA software handles all storage-related functions for guest VMs, which are also referred to as application VMs".

As examined in (A), the server obviously includes at least one processor. The "distributed virtual array (DVA) software" in the Cited Invention is obviously software to be executed by the processor, and corresponds to the "distributed virtual array (DVA) sub-system (405) comprising computer-executable code" in the Amended Invention.

The "DVA software" in the Cited Invention, which "handles all storage-related functions", obviously handles storage-related function on data corresponding to a write request regarding the write operation examined in (E).

Therefore, the Amended Invention and the Cited Invention are identical in that "a host server" "includes" "a distributed virtual array (DVA) sub-system (405) comprising computer-executable code which, when executed on the at least one processor, causes at least one processor to execute at least one storage processing function on the data corresponding to the write request", although they have the different features described later.

(G) As examined in (E), the write processing represented by the description in the Cited Invention, "if the storage access operation (read or write operation normally) is a WRITE, data are written directly to the NVRAM of one or more of the storage nodes", is considered as the "first write processing". The "NVRAM of the storage nodes" in the Cited Invention corresponds to the "remote non-volatile memory" of the "storage nodes (310)" in the Amended Invention. The "driver" in the Cited Invention is considered as a kind of system software component. Thus, it can be said that the Cited Invention is configured so that the driver as a kind of system software component writes data to the remote non-volatile memory of the storage nodes in the first write processing.

Therefore, considering the examination in (E), the Amended Invention and the Cited Invention are identical in that the "system software (115) component" "in first write processing, writes data corresponding to the write requests from the host servers to at least one remote non-volatile memory device (330) of the storage nodes (310)", although they have the different features described later.

(H) As examined in (E), the write processing represented by the description in the Cited Invention, "then" "the Write data plus the erasure code block are written to one or more storage nodes", is considered as the" second write processing". It should be said that the description in the Cited Invention, "an erasure code block is computed for the stripe of new Write data", means that computation of an erasure code block which is a kind of storage processing function is performed on data corresponding to a write request. The Cited Invention is configured so that "the erasure code block is written to one or more storage nodes". The "erasure code block" is considered as data generated by a storage processing function.

Therefore, considering the examination in (E), the Amended Invention and the Cited Invention are identical in that "... in second write processing, writes data generated by the at least one storage processing function on the data corresponding to the write request to the storage nodes (310)", although they have the different features as described later.

(I) The Cited Invention is configured so that "new Write data are stored to the NVRAM of at least one of the storage nodes so that if there is a power failure, a copy of the new Write data survives". Thus, when a power failure occurs, the "NVRAM" is considered to function as a temporary backup data storage.

Therefore, considering the examination in (G), the Amended Invention and the Cited Invention are identical in that "at least one remote non-volatile memory device (330) constitutes a temporary backup data storage system".

(J) As examined in (H), it should be said that the description in the Cited Invention, "an erasure code block is computed for the stripe of new Write data", means that computation of an erasure code block which is a kind of storage processing functions is performed on data corresponding to a write request. It is obvious for a person skilled in the art that error-correction data being a parity are computed over the data to be written in computation of an erasure code block. The Cited Invention is configured so that "the Write data plus the erasure code block are written to one or more storage nodes". Thus, it can be said that the Cited Invention performs "computing error-correction data over the data to be written and storing both the data to be written and the computed error-correction in at least one of the storage nodes" as a "storage processing function".

Therefore, the Amended Invention and the Cited Invention are identical in "including" "computing error-correction data over the data to be written and storing both the data to be written and the computed error-correction in at least one of the storage nodes (310)" as "at least one", regarding the following description:

"the at least one storage processing function includes at least one of:

computing a fingerprint of data written and storing the fingerprinted data in a selected one of the storage nodes (310);

compressing data to be written and sending the data in compressed form for storage in at least one of the storage nodes (310);

computing error-correction data over the data to be written and storing both the data to be written and the computed error-correction in at least one of the storage nodes (310); and

logging data writes and transmitting corresponding logging information to at least one node's storage controller (320-1, 320-2, ..., 320-N)."

B According to (A) to (J), the Amended Invention and the Cited Invention have the following corresponding feature and different features.

<Corresponding Feature>

"A data storage system comprising:

a plurality of host servers (100), each of which includes at least one processor (111) and which communicates with a plurality of storage nodes (310) over at least one network (200),

at least one storage node including at least one persistent storage device (340-1, 340-N) and at least one storage controller (320-1, 320-2, ... , 320-N),

the host servers including:

at least one process that issues data storage read/write requests;

a memory device forming a cache (113) configured for caching data corresponding to a write request in a non-persistent manner,

a system software (115) component configured to write data corresponding to the write request at least twice; and

a distributed virtual array (DVA) sub-system (405) comprising computer-executable code which, when executed on the at least one processor, causes at least one processor to execute at least one storage processing function on the data corresponding to the write request, wherein

the system software (115) component,

in first write processing, writes data corresponding to the write requests from the host servers to at least one remote non-volatile memory device (330) of the storage nodes (310),

in second write processing, writes data generated by the at least one storage processing function on the data corresponding to the write request to the storage nodes (310),

the at least one remote non-volatile memory device (330) constitutes a temporary backup data storage system,

the at least one storage processing function includes

computing error-correction data over the data to be written and storing both the data to be written and the computed error-correction in at least one of the storage nodes (310)."

<Different Feature 1>

The Amended Invention is configured to perform first write processing "without involving any of the persistent storage devices (340-1, 340-N) of the storage nodes (310)". The Cited Invention does not include such specification.

<Different Feature 2>

The Amended Invention is configured to perform second write processing "without involving the at least one remote non-volatile memory device (330)", and to "write data to the at least one persistent storage devices (340-1, 340-N)". The Cited Invention does not include such specification.

(4) Judgment by the body

The above-mentioned different features are examined.

A　Regarding Different Feature 1

As examined in "(3) A" (B) and (G), the "NVRAM" and the "disks" in the Cited Invention correspond to the "remote non-volatile memory" and the "persistent storage devices" in the Amended Invention, respectively.

The Cited Invention is configured so that "data are written directly to the NVRAM of one or more of the storage nodes" in the first write processing. Thus, as long as data are "directly written to the NVRAM" of the storage nodes, there is no need to access the "multiple disks" of the storage nodes, obviously.

Accordingly, the matter in the Cited Invention of performing first write processing "without involving any of the persistent storage devices (340-1, 340-N) of the storage nodes (310)" can be easily implemented by a person skilled in the art as long as the first write processing is to "write data directly to the NVRAM".

B　Regarding Different Feature 2

The Cited Invention is configured so that "the Write data plus the erasure code block are written to one or more storage nodes", and the storage nodes in the Cited Invention include "disks" which are persistent storage devices. The Cited Invention is a "distributed virtual array data storage system". Storage systems are to store data persistently. Thus, it is reasonable to understand that the second write processing in which "the Write data plus the erasure code block are written to one or more storage nodes" means that data are written to "disks" which are persistent storage devices of the storage nodes. There are no technical grounds for indispensably requiring access to the "NVRAM" of the storage nodes in writing "the Write data plus the erasure code block" to the "disks" of the storage nodes.

Accordingly, the matter in the Cited Invention of performing the second write processing "without involving the at least one remote non-volatile memory device (330)", and "writing data to the at least one persistent storage devices (340-1, 340-N)" of the storage nodes (310) can be easily implemented by a person skilled in the art, according to the facts that the Cited Invention is a storage system including disks and that there are no technical grounds for indispensably requiring access to the "NVRAM" in writing data to the disks.

C　Summary

As examined in A and B, configurations relating to the Different Feature 1 and the Different Feature 2 could be easily conceived by a person skilled in the art on the basis of the technology described in the Cited Invention. Even after comprehensively considering the different features, an advantageous effect to be produced by the Amended Invention is only within a scope predicted from an advantageous effect to be produced by the technology described in the Cited Invention, and it cannot be particularly significant.

Therefore, the Amended Invention could have been easily made by a person skilled in the art on the basis of the technical matters described in the Cited Invention. The Appellant cannot be granted a patent independently at the time of patent application under the provisions of Article 29(2) of the Patent Act.

3 Closing on the Amendment

The Amendment violates the provisions of Article 126(7) of the Patent Act which is applied mutatis mutandis pursuant to Article 17-2(6) of the Patent Act, and should be dismissed under the provisions of Article 53(1) of the Patent Act which is applied mutatis mutandis pursuant to the provisions of Article 159(1) of the Patent Act.

Therefore, the decision is made in accordance with Conclusion of Decision to Dismiss Amendment.

No. 3 Regarding the Invention

1 Recognition of the Invention

The written amendment submitted on February 19, 2020 was dismissed as above. The invention according to the claims of the present application is specified by the matters recited in Claims 1 to 23 of the scope of claims amended by the written amendment submitted on June 7, 2019. The invention according to Claim 1 (hereinafter referred to as "the Invention") is as follows, which is described in "No. 2 [Reason] 1 (2)" and specified by the matters recited in Claim 1 of the invention.

"A data storage system comprising:

a plurality of host servers (100), each of which includes at least one processor (111) and which communicates with a plurality of storage nodes (310) over at least one network (200),

at least one storage node including at least one persistent storage device (340-1, 340-N) and at least one storage controller (320-1, 320-2, ... , 320-N), the host servers including:

at least one process that issues data storage read/write requests;

a memory device forming a cache (113) configured for caching data corresponding to a write request in a non-persistent manner,

a system software (115) component configured to write data corresponding to the write request to at least one remote non-volatile memory device (330) as a temporary storage device; and

a distributed virtual array (DVA) sub-system (405) comprising computer-executable code which, when executed on the at least one processor, causes at least one processor to execute at least one storage processing function on the data corresponding to the write request, wherein

the at least one storage processing function includes at least one of:

computing a fingerprint of data written and storing the fingerprinted data in a selected one of the storage nodes (310);

compressing data to be written and sending the data in compressed form for storage in at least one of the storage nodes (310);

computing error-correction data over the data to be written and storing both the data to be written and the computed error-correction in at least one of the storage nodes (310); and

logging data writes and transmitting corresponding logging information to at least one node's storage controller (320-1, 320-2, ..., 320-N)."

2 Reasons for refusal stated in the examiner's decision

The reasons for refusal stated in the examiner's decision is as follows: The invention according to Claim 1 of this application could have been easily made by a person ordinarily skilled in the art of the invention before the filing of the application, on the basis of the technology described in the following Cited Document 1 which was distributed or made publicly available through an electric telecommunication line prior to the filing of the application. Thus, the Appellant should not be granted a patent for the invention under the provisions of Article 29(2) of the Patent Act.

1. Specification of U. S. Patent Application Publication No. 2015/0127975

3 Cited Document

Cited Document 1 and the matters described therein cited in the reasons for refusal stated in the examiner's decision are as described in "No. 2 [Reason] 2 (2)"

4 Comparison / judgment

The Invention is to delete, regarding the write processing performed by the "software (115) component" in the Amended Invention examined in "No. 2 [Reason] (1)", the following limitations:

"at least twice" for write processing, and

"in first write processing, writes data corresponding to the write requests from the host servers to at least one remote non-volatile memory device (330) of the storage nodes (310) without involving any of the persistent storage devices (340-1, 340-N) of the storage nodes (310),

in second write processing, writes data generated by the at least one storage processing function on the data corresponding to the write request to the at least one persistent storage device (340-1, 340-N) of the storage nodes (310) without involving the at least one remote non-volatile memory device (330),

the at least one remote non-volatile memory device (330) constitutes a temporary backup data storage system".

Accordingly, the Amended Invention corresponding to an invention including all of the matters specifying the Invention and other matters added thereto could have been easily made by a person skilled in the art on the basis of the technology described in the Cited Invention as described in "No. 2 [Reason] 2 (3), (4)". Thus, the Invention also could have been easily made by a person skilled in the art on the basis of the technology described in the Cited Invention.

No. 4 Closing

As described above, the Appellant cannot be granted a patent for the invention according to Claim 1 of the present application under the provisions of Article 29(2) of the Patent Act. The present application should be rejected without examining inventions according to other claims.

Therefore, the appeal decision shall be made as described in the conclusion.

November 11, 2020

Chief administrative judge: TANAKA, Hideto
Administrative judge: TSUKINO, Yoichiro
Administrative judge: YAMASAWA, Hiroshi