



(12) **Veröffentlichung**

der internationalen Anmeldung mit der
 (87) Veröffentlichungs-Nr.: **WO 2021/005444**
 in der deutschen Übersetzung (Art. III § 8 Abs. 2
 IntPatÜbkG)
 (21) Deutsches Aktenzeichen: **11 2020 002 834.8**
 (86) PCT-Aktenzeichen: **PCT/IB2020/056004**
 (86) PCT-Anmeldetag: **25.06.2020**
 (87) PCT-Veröffentlichungstag: **14.01.2021**
 (43) Veröffentlichungstag der PCT Anmeldung
 in deutscher Übersetzung: **17.03.2022**

(51) Int Cl.: **G06F 16/13 (2019.01)**
G06F 13/42 (2006.01)

(30) Unionspriorität:
16/505,453 **08.07.2019** **US**

(71) Anmelder:
International Business Machines Corporation,
Armonk, NY, US

(74) Vertreter:
Richardt Patentanwälte PartG mbB, 65185
Wiesbaden, DE

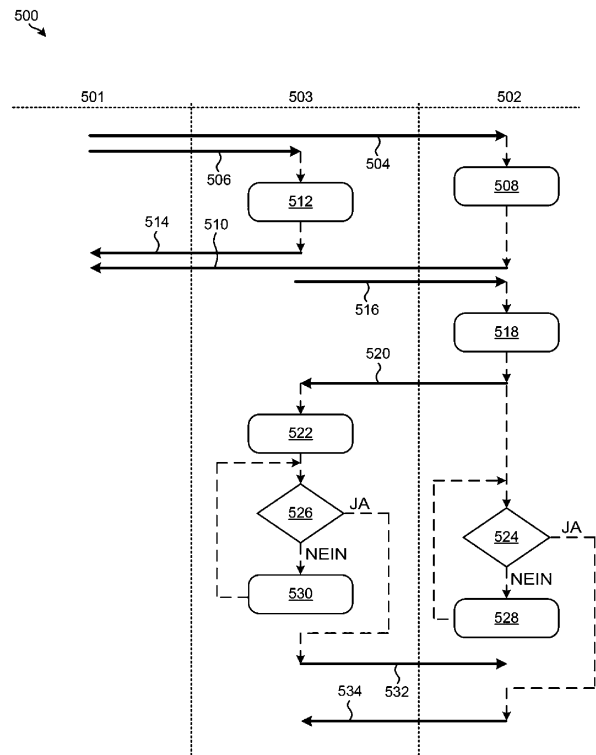
(72) Erfinder:
Yudenfriend, Harry, Poughkeepsie, NY, US;
Sutton, Peter Grimm, Poughkeepsie, NY, US;
Compton, Scott, Poughkeepsie, NY, US

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **SPIEGELN VON SCHREIBOPERATIONEN ZWISCHEN DATENSPEICHEREINHEITEN**

(57) Zusammenfassung: Ein durch einen Computer implementiertes Verfahren gemäß einem Ansatz umfasst: Empfangen einer ersten Anforderung zum Durchführen einer Schreiboperation von einem Host und Durchführen der Schreiboperation. Metadaten, die der ersten Anforderung entsprechen, werden an die sekundäre Datenspeichereinheit gesendet, und Metadaten, die einer zweiten Anforderung zum Durchführen der Schreiboperation entsprechen, werden von der sekundären Datenspeichereinheit empfangen, wobei die zweite Anforderung vom Host in der sekundären Datenspeichereinheit empfangen wurde. Die Metadaten, die der ersten und zweiten Anforderung entsprechen, werden verwendet, um zu ermitteln, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde. Als Reaktion auf ein Feststellen, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit gespiegelt wurde, wird eine Antwort an die sekundäre Datenspeichereinheit gesendet, die anzeigt, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurde.



Beschreibung

TECHNISCHES GEBIET

[0001] Die vorliegende Erfindung bezieht sich auf Datenspeichersysteme und insbesondere auf Spiegeln von Schreiboperationen zwischen mehr als einer Datenspeichereinheit.

HINTERGRUND

[0002] In herkömmlichen Datenreplikationssystemen senden Benutzer Eingabe/Ausgabe-(E/A-)Anforderungen an einen einzelnen Speicherort, z.B. an einen primären Speicherort, der Daten schließlich entsprechend den empfangenen E/A-Anforderungen ändert. Um die Datenspeicherung zu verbessern, implementieren herkömmliche Datenreplikationssysteme manchmal auch einen zweiten Speicherort, in dem eine zweite Kopie (z.B. eine Sicherungskopie) der am primären Speicherort gespeicherten Daten abgelegt wird. Diese redundante Kopie der Daten am Wiederherstellungsspeicherort ist besonders nützlich in Situationen, in denen der primäre Speicherort nicht mehr verfügbar ist und/oder Datenverluste auftreten. In einigen Fällen kann der Wiederherstellungsspeicherort sogar die Zuständigkeit für den Betrieb übernehmen, wenn festgestellt wird, dass der primäre Speicherort dazu nicht in der Lage ist.

[0003] In solchen herkömmlichen Systemen leitet der primäre Speicherort die von den Benutzern empfangenen E/A-Anforderungen zum Implementieren an den zweiten Speicherort weiter. Die Datenspeicherung wird dadurch zwar verbessert, doch durch Erfüllen der zusätzlichen Anforderungen erhöht sich der Verarbeitungsaufwand dieses Datenspeicherungsschemas beträchtlich. So werden beispielsweise E/A-Anforderungen zwischen den einzelnen Speicherorten übertragen, was zu Verzögerungen in der Durchführung führt, die mit der Anzahl der implementierten Speicherorte zunehmen. Darüber hinaus unterbricht ein Fehlerereignis in einem der Speicherorte und/oder den dazwischen bestehenden Verbindungen die Übertragung von E/A-Anforderungen zwischen diesen und führt dazu, dass mehr als eine Kopie der Daten nicht mehr synchronisiert ist. Dies wiederum muss behoben werden, bevor das System betriebsbereit ist, was zu zusätzlichen Verzögerungen in der Durchführung führt.

KURZDARSTELLUNG

[0004] Ein durch einen Computer implementiertes Verfahren gemäß einem Aspekt der Erfindung dient zum Spiegeln von Schreiboperationen zwischen primärer und sekundärer Datenspeichereinheit. Das durch einen Computer implementierte Verfahren umfasst: Empfangen einer ersten Anforderung zum

Durchführen einer Schreiboperation von einem Host und Durchführen der Schreiboperation. Metadaten, die der ersten Anforderung entsprechen, werden an die sekundäre Datenspeichereinheit gesendet, und Metadaten, die einer zweiten Anforderung zum Durchführen der Schreiboperation entsprechen, werden ebenfalls von der sekundären Datenspeichereinheit empfangen. Daraus folgt, dass die zweite Anforderung in der sekundären Datenspeichereinheit vom Host empfangen wurde. Die Metadaten, die der ersten und zweiten Anforderung entsprechen, werden verwendet, um zu ermitteln, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde. Als Reaktion auf ein Feststellen, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit gespiegelt wurde, wird eine Antwort an die sekundäre Datenspeichereinheit gesendet, die anzeigt, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurde.

[0005] Infolgedessen werden die Durchführungszeiten verkürzt, da der Host parallel mit der primären und sekundären Datenspeichereinheit Daten austauschen kann. Verschiedene der hier beschriebenen Ansätze sind beispielsweise in der Lage, den Gesamtaufwand zu verringern. Ebenso können die primäre und die sekundäre Datenspeichereinheit parallel arbeiten und zwischen diesen Synchronisationsvorgänge durchführen, ohne auf Eingaben des Hosts angewiesen zu sein. Auf diese Weise können die Datenspeichereinheiten dazu dienen, dass die Daten entsprechend den vorgesehenen Datenspeicherschemata weitergegeben werden. Dadurch werden der Netzwerkverkehr verringert und die Datenspeicherung verbessert, da aktiv sichergestellt wird, dass Daten zwischen mehreren Datenspeichereinheiten gespiegelt werden. Entsprechend ist es vorzuziehen, dass die verschiedenen Prozesse in diesen Ansätzen auf alle Schreiboperationen (z.B. alle Anwendungsschreiboperationen) angewendet werden, die vom Host empfangen werden.

[0006] In einigen Ausführungsformen umfasst das durch einen Computer implementierte Verfahren außerdem: Empfangen einer Leseanforderung und Ermitteln, ob in der Leseanforderung angegebene Daten zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden. Als Reaktion auf ein Feststellen, dass die in der Leseanforderung angegebenen Daten nicht zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden, wird ein Durchführen der Leseanforderung verzögert. Als Reaktion auf ein Feststellen, dass die in der Leseanforderung angegebenen Daten zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden, wird die Leseoperation jedoch durchgeführt.

[0007] Auf diese Weise wird die Betriebseffizienz des übergeordneten Systems weiterhin verbessert, indem Situationen weitgehend vermieden werden, in denen eine Leseoperation versucht wird, nur um festzustellen, dass die Daten, die gelesen werden sollen, noch nicht im Speicher gespeichert sind, noch nicht in Übereinstimmung mit einer geänderten Schreiboperation aktualisiert wurden usw. Außerdem können diese Operationen implementiert werden, ohne das Durchführen zu unterbrechen. Dies ist besonders wünschenswert, da die mit den vorstehend beschriebenen Ansätzen erzielten Verringerungen der Verzögerung und die Datenspeicherung mit den durch den vorliegenden Ansatz erzielten Verbesserungen beim Durchführen der Leseoperation kombiniert werden können.

[0008] Ein Computerprogrammprodukt gemäß einem weiteren Aspekt der Erfindung dient zum Spiegeln von Schreiboperationen zwischen primärer und sekundärer Datenspeichereinheit. Das Computerprogrammprodukt umfasst ein durch einen Computer lesbares Speichermedium mit darauf enthaltenen Programmanweisungen. Außerdem sind die Programmanweisungen von einem Prozessor lesbar und/oder ausführbar, um den Prozessor zu veranlassen, das vorgenannte Verfahren durchzuführen.

[0009] Eine primäre Datenspeichereinheit umfasst gemäß einem weiteren Aspekt: einen Prozessor und eine in den Prozessor integrierte, vom Prozessor ausführbare oder in den Prozessor integrierte und von diesem ausführbare Logik. Die Logik ist konfiguriert, um: das vorgenannte Verfahren durchzuführen.

[0010] Ein durch einen Computer implementiertes Verfahren gemäß einem weiteren Aspekt dient zum Spiegeln von Schreiboperationen zwischen primärer und sekundärer Datenspeichereinheit. Das durch einen Computer implementierte Verfahren umfasst: Empfangen einer ersten Anforderung zum Durchführen einer Schreiboperation von einem Host und Durchführen der Schreiboperation. Metadaten, die der ersten Anforderung entsprechen, werden an die primäre Datenspeichereinheit gesendet. Metadaten, die einer zweiten Anforderung zum Durchführen der Schreiboperation entsprechen, werden ebenfalls von der primären Datenspeichereinheit empfangen, wobei die zweite Anforderung vom Host in der primären Datenspeichereinheit empfangen wurde. Die Metadaten, die der ersten und zweiten Anforderung entsprechen, werden verwendet, um zu ermitteln, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde. Als Reaktion auf ein Feststellen, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit gespiegelt wurde, wird weiterhin eine Antwort an die primäre Datenspeichereinheit gesendet, die anzeigt, dass die Schreiboperation

zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurde.

[0011] Ein durch einen Computer implementiertes Verfahren gemäß einem weiteren Aspekt dient zum Spiegeln von Schreiboperationen zwischen primärer und sekundärer Datenspeichereinheit. Hier umfasst das durch einen Computer implementierte Verfahren: Senden einer ersten Anforderung zum Durchführen einer Schreiboperation an eine primäre Datenspeichereinheit und Senden einer zweiten Anforderung zum Durchführen der Schreiboperation an eine sekundäre Datenspeichereinheit. Eine erste Antwort wird von der primären Datenspeichereinheit empfangen, die anzeigt, dass die Schreiboperation durchgeführt wurde. Darüber hinaus wird eine erste Antwort von der primären Datenspeichereinheit empfangen, die anzeigt, dass die Schreiboperation durchgeführt wurde.

[0012] Wie oben beschrieben, werden die Durchführungszeiten verkürzt, da der Host parallel mit der primären und sekundären Datenspeichereinheit Daten austauschen kann. Verschiedene der hier beschriebenen Ansätze sind beispielsweise in der Lage, den Gesamtaufwand zu verringern. Ebenso können die primäre und die sekundäre Datenspeichereinheit parallel arbeiten und zwischen diesen Synchronisationsvorgänge durchführen, ohne auf Eingaben des Hosts angewiesen zu sein. Auf diese Weise können die Datenspeichereinheiten dazu dienen, dass die Daten entsprechend den vorgesehenen Datenspeicherschemata weitergegeben werden. Dadurch werden der Netzwerkverkehr verringert und die Datenspeicherung verbessert, da aktiv sichergestellt wird, dass Daten zwischen mehreren Datenspeichereinheiten gespiegelt werden. Entsprechend ist es vorzuziehen, dass die verschiedenen Prozesse in diesen Ansätzen auf alle Schreiboperationen (z.B. alle Anwendungsschreiboperationen) angewendet werden, die vom Host empfangen werden.

[0013] Weitere Aspekte und Ansätze der vorliegenden Erfindung ergeben sich aus der folgenden ausführlichen Beschreibung, die in Verbindung mit den Zeichnungen die Grundgedanken der Erfindung beispielhaft veranschaulicht.

Figurenliste

Fig. 1 ist eine Netzwerkarchitektur gemäß einem Ansatz.

Fig. 2 ist eine repräsentative Hardwareumgebung, die den Servern und/oder Clients von **Fig. 1** gemäß einem Ansatz zugehörig sein kann.

Fig. 3 ist ein Datenspeichersystem mit mehreren Ebenen gemäß einem Ansatz.

Fig. 4 ist eine Darstellung einer Teilansicht eines verteilten Datenspeichersystems gemäß einem Ansatz.

Fig. 5A ist ein Ablaufplan für ein Verfahren gemäß einem Ansatz.

Fig. 5B ist ein Ablaufplan für ein Verfahren gemäß einem Ansatz.

AUSFÜHRLICHE BESCHREIBUNG

[0014] Die folgende Beschreibung dient der Veranschaulichung der allgemeinen Grundgedanken der vorliegenden Erfindung und soll die hier beanspruchten Aspekte der Erfindung nicht einschränken. Darüber hinaus können hier beschriebene besondere Merkmale in Verbindung mit anderen beschriebenen Merkmalen in jeder der verschiedenen möglichen Kombinationen und Permutationen verwendet werden.

[0015] Sofern hier nicht ausdrücklich anders definiert, sind alle Begriffe so weit wie möglich auszulegen, einschließlich der aus der Beschreibung implizierten Bedeutungen, sowie der Bedeutungen, die von Fachleuten verstanden werden und/oder die in Wörterbüchern, Abhandlungen usw. definiert sind.

[0016] Sofern nicht anders angegeben, ist ferner darauf hinzuweisen, dass die Singularformen „ein/eine/einer/eines“ und „der/die/das“, wie sie in der Beschreibung und den beigefügten Ansprüchen verwendet werden, auch die Pluralform enthalten. Es versteht sich ferner, dass die Begriffe „aufweisen“ und/oder „aufweisend“, wenn sie in dieser Beschreibung verwendet werden, die Anwesenheit von angegebenen Merkmalen, Ganzzahlen, Schritten, Operationen, Elementen und/oder Komponenten spezifizieren, jedoch nicht die Anwesenheit oder Hinzufügung von einem oder mehreren anderen Merkmalen, Ganzzahlen, Schritten, Operationen, Elementen, Komponenten und/oder Gruppen davon ausschließen.

[0017] Die folgende Beschreibung offenbart mehrere bevorzugte Ansätze von Systemen, Verfahren und Computerprogrammprodukten zum Spiegeln von Schreiboperationen zwischen mehr als einer Speichereinheit. Es sei darauf hingewiesen, dass sich „Spiegeln“ von Schreiboperationen zwischen Speichereinheiten auf den Prozess zum Abgleichen von Unterschieden zwischen Kopien derselben Daten an zwei verschiedenen Speicherorten bezieht, wie es für den Fachmann beim Lesen der vorliegenden Beschreibung offensichtlich ist. Entsprechend sind verschiedene der hier beschriebenen Ansätze wünschenswerterweise in der Lage, die Durchführungszeiten zu verkürzen, da der Host parallel mit den mehreren Datenspeichereinheiten Daten austauschen kann. Ebenso können die Datenspeiche-

reinheiten parallel arbeiten und zwischen diesen Synchronisationsvorgänge durchführen, ohne auf Eingaben des Hosts angewiesen zu sein. Auf diese Weise können die Steuereinheiten in jeder der Datenspeichereinheiten dazu dienen, dass die Daten entsprechend den vorgesehenen Datenspeicherschemata weitergegeben werden, wodurch erhebliche Verbesserungen im Vergleich zu herkömmlichen Verfahren erzielt werden, wie z.B. nachfolgend ausführlicher beschrieben wird.

[0018] In einem allgemeinen Ansatz wird ein durch einen Computer implementiertes Verfahren verwendet, um Schreiboperationen zwischen primärer und sekundärer Datenspeichereinheit zu spiegeln. Das durch einen Computer implementierte Verfahren umfasst: Empfangen einer ersten Anforderung zum Durchführen einer Schreiboperation von einem Host und Durchführen der Schreiboperation. Metadaten, die der ersten Anforderung entsprechen, werden an die sekundäre Datenspeichereinheit gesendet, und Metadaten, die einer zweiten Anforderung zum Durchführen der Schreiboperation entsprechen, werden ebenfalls von der sekundären Datenspeichereinheit empfangen. Daraus folgt, dass die zweite Anforderung in der sekundären Datenspeichereinheit vom Host empfangen wurde. Die Metadaten, die der ersten und zweiten Anforderung entsprechen, werden verwendet, um zu ermitteln, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde. Als Reaktion auf ein Feststellen, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit gespiegelt wurde, wird eine Antwort an die sekundäre Datenspeichereinheit gesendet, die anzeigt, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurde.

[0019] In einem weiteren allgemeinen Ansatz wird ein Computerprogrammprodukt verwendet, um Schreiboperationen zwischen primärer und sekundärer Datenspeichereinheit zu spiegeln. Das Computerprogrammprodukt umfasst ein durch einen Computer lesbares Speichermedium mit darauf enthaltenen Programmanweisungen. Außerdem sind die Programmanweisungen von einem Prozessor lesbar und/oder ausführbar, um den Prozessor zu veranlassen, das vorgenannte Verfahren durchzuführen.

[0020] In einem weiteren allgemeinen Ansatz umfasst eine primäre Datenspeichereinheit: einen Prozessor und eine in den Prozessor integrierte, vom Prozessor ausführbare oder in den Prozessor integrierte und von diesem ausführbare Logik. Die Logik ist konfiguriert, um: das vorgenannte Verfahren durchzuführen.

[0021] In einem weiteren allgemeinen Ansatz wird ein durch einen Computer implementiertes Verfah-

ren verwendet, um Schreiboperationen zwischen primärer und sekundärer Datenspeichereinheit zu spiegeln. Das durch einen Computer implementierte Verfahren umfasst: Empfangen einer ersten Anforderung zum Durchführen einer Schreiboperation von einem Host und Durchführen der Schreiboperation. Metadaten, die der ersten Anforderung entsprechen, werden an die primäre Datenspeichereinheit gesendet. Metadaten, die einer zweiten Anforderung zum Durchführen der Schreiboperation entsprechen, werden ebenfalls von der primären Datenspeichereinheit empfangen, wobei die zweite Anforderung vom Host in der primären Datenspeichereinheit empfangen wurde. Die Metadaten, die der ersten und zweiten Anforderung entsprechen, werden verwendet, um zu ermitteln, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde. Als Reaktion auf ein Feststellen, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit gespiegelt wurde, wird weiterhin eine Antwort an die primäre Datenspeichereinheit gesendet, die anzeigt, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurde.

[0022] In einem weiteren allgemeinen Ansatz wird ein durch einen Computer implementiertes Verfahren verwendet, um Schreiboperationen zwischen primärer und sekundärer Datenspeichereinheit zu spiegeln. Hier umfasst das durch einen Computer implementierte Verfahren: Senden einer ersten Anforderung zum Durchführen einer Schreiboperation an eine primäre Datenspeichereinheit und Senden einer zweiten Anforderung zum Durchführen der Schreiboperation an eine sekundäre Datenspeichereinheit. Eine erste Antwort wird von der primären Datenspeichereinheit empfangen, die anzeigt, dass die Schreiboperation durchgeführt wurde. Darüber hinaus wird eine erste Antwort von der primären Datenspeichereinheit empfangen, die anzeigt, dass die Schreiboperation durchgeführt wurde.

[0023] **Fig. 1** veranschaulicht eine Architektur 100 gemäß einem Ansatz. Wie in **Fig. 1** dargestellt, wird eine Mehrzahl von entfernt angeordneten Netzwerken 102 bereitgestellt, die ein erstes entfernt angeordnetes Netzwerk 104 und ein zweites entfernt angeordnetes Netzwerk 106 umfassen. Ein Gateway 101 kann mit dem entfernt angeordneten Netzwerken 102 und einem nahen Netzwerk 108 verbunden sein. Im Rahmen der vorliegenden Architektur 100 können die Netzwerke 104, 106 jeweils die Form eines lokalen Netzwerks (LAN) oder eines Weitbereichsnetzwerks (WAN) wie das Internet, eines öffentlichen Telefonnetzes (PSTN), eines internen Telefonnetzes usw. haben, ohne auf diese beschränkt zu sein.

[0024] Im Einsatz dient das Gateway 101 als Zugangspunkt von den entfernt angeordneten Netzwerken 102 zum nahen Netzwerk 108. Als solches kann das Gateway 101 als Router fungieren, der in der Lage ist, ein bestimmtes Datenpaket, das am Gateway 101 ankommt, weiterzuleiten, sowie als Switch, die den tatsächlichen Pfad in das und aus dem Gateway 101 für ein bestimmtes Paket bereitstellt.

[0025] Weiterhin ist mindestens ein Datenserver 114 enthalten, der mit dem nahen Netzwerk 108 verbunden ist und auf den über das Gateway 101 von den entfernt angeordneten Netzwerken 102 zugegriffen werden kann. Es ist zu beachten, dass der/die Datenserver 114 jede Art von Datenverarbeitungseinheit/Groupware enthalten kann/können. Mit jedem Datenserver 114 ist eine Mehrzahl von Benutzereinheiten 116 verbunden. Die Benutzereinheiten 116 können auch direkt über eines der Netzwerke 104, 106, 108 verbunden werden. Je nach Ansatz können solche Benutzereinheiten 116 einen Großrechner (z.B. wie hier beschrieben), einen Desktop-Computer, einen Laptop-Computer, einen Handheld-Computer, einen Drucker oder jede beliebige andere Art von Logik umfassen. Es ist zu beachten, dass in einem Ansatz eine Benutzereinheit 111 auch direkt mit jedem der Netzwerke verbunden sein kann.

[0026] Eine Peripherieeinheit 120 oder eine Reihe von Peripherieeinheiten 120, z.B. Faxgeräte, Drucker, vernetzte und/oder lokale Speichereinheiten oder-systeme usw., kann mit einem oder mehreren der Netzwerke 104, 106, 108 verbunden werden. Es ist zu beachten, dass Datenbanken und/oder zusätzliche Komponenten mit jeder Art von Netzwerkelement, das mit den Netzwerken 104, 106, 108 verbunden ist, verwendet oder in dieses integriert werden können. Im Zusammenhang mit der vorliegenden Beschreibung kann sich ein Netzwerkelement auf eine beliebige Komponente eines Netzwerkes beziehen.

[0027] Gemäß einigen Ansätzen können die hier beschriebenen Verfahren und Systeme mit und/oder auf virtuellen Systemen und/oder Systemen implementiert werden, die ein oder mehrere andere Systeme emulieren, wie z.B. ein UNIX-System, das eine IBM z/OS-Umgebung emuliert, ein UNIX-System, das eine MICROSOFT WINDOWS-Umgebung virtuell hostet, ein MICROSOFT WINDOWS-System, das eine IBM z/OS-Umgebung emuliert, usw. Diese Virtualisierung und/oder Emulation kann mithilfe von VMWARE-Software in einigen Ansätzen verbessert werden.

[0028] Bei weiteren Ansätzen können ein oder mehrere Netzwerke 104, 106, 108 eine Gruppe von Systemen darstellen, die allgemein als „Cloud“ bezeichnet werden. Beim Cloud-Computing werden gemein-

sam genutzte Ressourcen wie Rechenleistung, Peripherieeinheiten, Software, Daten, Server usw. jedem System in der Cloud in einer bedarfsorientierten Beziehung bereitgestellt und ermöglichen so den Zugriff auf und die Verteilung von Diensten über viele Datenverarbeitungssysteme hinweg. Beim Cloud-Computing handelt es sich in der Regel um eine Internetverbindung zwischen den in der Cloud arbeitenden Systemen, es können jedoch auch andere Verfahren zum Verbinden der Systeme verwendet werden.

[0029] Fig. 2 zeigt eine repräsentative Hardwareumgebung, die einer Benutzereinheit 116 und/oder einem Server 114 von Fig. 1 gemäß einem Ansatz zugehörig ist. Diese Figur zeigt eine typische Hardwarekonfiguration einer Workstation mit einer zentralen Verarbeitungseinheit 210, z.B. einem Mikroprozessor, und einer Reihe anderer Einheiten, die über einen Systembus 212 miteinander verbunden sind.

[0030] Die in Fig. 2 dargestellte Workstation umfasst einen Direktzugriffsspeicher (RAM) 214, einen Nur-Lese-Speicher (ROM) 216, einen Eingabe/Ausgabe-Adapter 218 zum Verbinden von Peripherieeinheiten wie z.B. Plattenspeichereinheiten 220 mit dem Bus 212, einen Benutzerschnittstellenadapter 222 zum Verbinden einer Tastatur 224, einer Maus 226, eines Lautsprechers 228, eines Mikrofons 232 und/oder anderer Benutzer-Schnittstelleneinheiten wie z.B. eines Touchscreens und einer Digitalkamera (nicht dargestellt) mit dem Bus 212, einen Datenübertragungsadapter 234 zum Verbinden der Workstation mit einem Datenübertragungsnetzwerk 235 (z.B. ein Datenverarbeitungsnetzwerk) und einen Anzeigeadapter 236 zum Verbinden des Buses 212 mit einer Anzeigeeinheit 238. Gemäß einem beispielhaften Ansatz, der die Erfindung in keiner Weise einschränken soll, können die Plattenspeichereinheiten 220 in einen DS8000 Plattenspeicher eingebaut werden, der von IBM mit einem Vertriebsbüro in 1 New Orchard Rd., Armonk, New York 10504 angeboten wird.

[0031] Auf der Workstation kann ein Betriebssystem, z.B. das Microsoft Windows® Operating System, ein MAC-Betriebssystem, ein UNIX-Betriebssystem usw., installiert sein. Es sei darauf hingewiesen, dass ein bevorzugter Ansatz auch auf anderen Plattformen und Betriebssystemen als den genannten implementiert werden kann. Ein bevorzugter Ansatz kann mit Assemblersprachen, C und/oder C++ oder anderen Programmiersprachen zusammen mit einer objektorientierten Programmiermethodik geschrieben werden. Objektorientierte Programmierung (OOP), die zunehmend zum Entwickeln komplexer Anwendungen eingesetzt wird, kann verwendet werden.

[0032] Mit Bezug nunmehr auf Fig. 3 wird ein Speichersystem 300 gemäß einem Ansatz dargestellt. Es ist zu beachten, dass einige der in Fig. 3 gezeigten Elemente als Hardware und/oder Software gemäß verschiedener Ansätze implementiert werden können. Das Speichersystem 300 kann einen Speichersystemmanager 312 für den Datenaustausch mit einer Mehrzahl von Medien und/oder Laufwerken auf mindestens einer höheren Speicherebene 302 und mindestens einer niedrigeren Speicherebene 306 enthalten. Die höhere(n) Speicherebene(n) 302 kann/können vorzugsweise ein oder mehrere Medien mit wahlfreiem Zugriff und/oder Direktzugriff 304 umfassen, wie z.B. Festplatten in Festplattenlaufwerken (hard disk drives, HDDs), nichtflüchtige Speicher (nonvolatile memory, NVM), Halbleiterspeicher in Halbleiterdatenträgern (solid state drives, SSDs), Flash-Speicher, SSD-Anordnungen, Flash-Speicher-Anordnungen usw. und/oder andere hier aufgeführte oder nach dem Stand der Technik bekannte Medien. Die niedrigere(n) Speicherebene(n) 306 kann/können vorzugsweise ein oder mehrere Speichermedien 308 mit geringerer Leistung umfassen, z.B. Medien mit sequenziellem Zugriff wie Magnetbänder in Bandlaufwerken und/oder optische Medien, langsamer zugreifende HDDs, langsamer zugreifende SSDs usw. und/oder andere hier aufgeführte oder nach dem Stand der Technik bekannte Medien. Eine oder mehrere zusätzliche Speicherebenen 316 können eine beliebige Kombination von Speichermedien enthalten, die von einem Entwickler des Systems 300 gewünscht wird. Außerdem kann jede der höheren Speicherebenen 302 und/oder der niedrigeren Speicherebenen 306 eine Kombination von Speichereinheiten und/oder Speichermedien enthalten.

[0033] Der Speichersystemmanager 312 kann mit den Laufwerken und/oder Speichermedien 304, 308 auf der/den höheren Speicherebene(n) 302 und der/den niedrigeren Speicherebene(n) 306 über ein Netzwerk 310, z.B. ein Speichernetzwerk (storage area network, SAN), wie in Fig. 3 dargestellt, oder über ein anderes geeignetes Netzwerk Daten austauschen. Der Speichersystemmanager 312 kann auch mit einem oder mehreren Host-Systemen (nicht dargestellt) über eine Host-Schnittstelle 314 Daten austauschen, die Teil des Speichersystemmanagers 312 sein kann. Der Speichersystemmanager 312 und/oder andere Komponenten des Speichersystems 300 kann/können in Hardware und/oder Software implementiert werden und einen (nicht dargestellten) Prozessor eines nach dem Stand der Technik bekannten Typs, wie z.B. eine Zentraleinheit (CPU), eine vor Ort programmierbare Gatter-Anordnung (field programmable gate array, FPGA), eine anwendungsspezifische integrierte Schaltung (application specific integrated circuit, ASIC) usw., zum Ausführen von Befehlen verwenden. Natürlich kann jede beliebige Anordnung eines Speichersystems

verwendet werden, wie es für den Fachmann beim Lesen der vorliegenden Beschreibung offensichtlich ist.

[0034] In mehreren Ansätzen kann das Speichersystem 300 eine beliebige Anzahl von Datenspeicherebenen umfassen und innerhalb jeder Speicherebene die gleichen oder unterschiedliche Speichermedien enthalten. Beispielsweise kann jede Datenschichtebene den gleichen Typ von Speichermedien, z.B. HDDs, SSDs, Medien mit sequenziellem Zugriff (Band in Bandlaufwerken, optische Platte in optischen Plattenlaufwerken usw.), Medien mit direktem Zugriff (CD-ROM, DVD-ROM usw.) oder eine beliebige Kombination von Medienspeichertypen umfassen. In einer solchen Konfiguration kann eine höhere Speicherebene 302 eine Mehrheit von SSD-Speichermedien zum Speichern von Daten in einer leistungsfähigeren Speicherumgebung umfassen, und die übrigen Speicherebenen, darunter die niedrigere Speicherebene 306 und zusätzliche Speicherebenen 316, können jede beliebige Kombination von SSDs, HDDs, Bandlaufwerken usw. zum Speichern von Daten in einer weniger leistungsfähigen Speicherumgebung umfassen. Auf diese Weise können Daten, auf die häufiger zugegriffen wird, Daten mit höherer Priorität, Daten, auf die schneller zugegriffen werden muss, usw. in der höheren Speicherebene 302 gespeichert werden, während Daten, die keines dieser Attribute besitzen, in den zusätzlichen Speicherebenen 316, darunter die niedrigere Speicherebene 306, gespeichert werden können. Natürlich kann ein Fachmann, wenn er die vorliegenden Beschreibungen liest, viele andere Kombinationen von Speichermedientypen entwickeln, die er in verschiedene Speicherschemata gemäß den hier vorgestellten Ansätzen implementieren kann.

[0035] Gemäß einigen Ansätzen kann das Speichersystem (z.B. 300) eine Logik enthalten, die so konfiguriert ist, dass sie eine Anforderung zum Öffnen eines Datensatzes empfängt, eine Logik, die so konfiguriert ist, dass sie ermittelt, ob der angeforderte Datensatz auf einer niedrigeren Speicherebene 306 eines Datenspeichersystems 300 mit mehreren Ebenen in mehreren zugehörigen Teilen gespeichert ist, eine Logik, die so konfiguriert ist, dass sie jeden zugehörigen Teil des angeforderten Datensatzes auf eine höhere Speicherebene 302 des Datenspeichersystems 300 mit mehreren Ebenen verschiebt, und eine Logik, die so konfiguriert ist, dass sie den angeforderten Datensatz auf der höheren Speicherebene 302 des Datenspeichersystems 300 mit mehreren Ebenen aus den zugehörigen Teilen zusammensetzt.

[0036] Natürlich kann diese Logik als Verfahren auf einer beliebigen Einheit und/oder einem beliebigem System oder als Computerprogrammprodukt gemäß verschiedenen Ansätzen implementiert werden.

[0037] Wie bereits erwähnt, werden mehrere Speichereinheiten implementiert, um eine oder mehrere redundante Kopien von Daten zu erhalten und die Datenspeicherung zu verbessern. Diese redundanten Kopien der Daten sind besonders nützlich in Situationen, in denen der primäre Speicherort nicht mehr verfügbar ist und/oder Datenverluste auftreten. In einigen Fällen können Wiederherstellungsspeicherorte die Zuständigkeit für den Betrieb übernehmen, wenn festgestellt wird, dass der primäre Speicherort dazu nicht in der Lage ist.

[0038] Herkömmliche Datenreplikationssysteme verzeichnen jedoch einige Leistungseinbußen beim Implementieren von E/A-Benutzeranforderungen zwischen mehreren Speichereinheiten. In solchen herkömmlichen Systemen leitet der primäre Speicherort die von den Benutzern empfangenen E/A-Anforderungen zum Implementieren an den zweiten Speicherort weiter. Die Datenspeicherung wird dadurch zwar tendenziell verbessert, doch durch Erfüllen der zusätzlichen Anforderungen erhöht sich der Verarbeitungsaufwand dieses Datenspeicherungsschemas beträchtlich. So werden beispielsweise E/A-Anforderungen zwischen den einzelnen Speicherorten übertragen, was zu Verzögerungen in der Durchführung führt, die mit der Anzahl der implementierten Speicherorte zunehmen.

[0039] Darüber hinaus unterbricht ein Fehlerereignis in einem der Speicherorte und/oder den dazwischen bestehenden Verbindungen die Übertragung von E/A-Anforderungen zwischen diesen und führt dazu, dass mehr als eine Kopie der Daten nicht mehr synchronisiert ist. Dies wiederum muss behoben werden, bevor das System wieder betriebsbereit ist, was zu zusätzlichen Verzögerungen in der Durchführung führt. Die Art und Weise, wie es zur fehlenden Synchronisation bei den Daten kommt, und der Bereich, auf den sich diese erstreckt, hängen jedoch davon ab, wann und wo das Fehlerereignis auftritt. Dies erhöht die Komplexität des Wiederherstellungsprozesses. So werden z.B. alle E/A-Anforderungen, die zum Zeitpunkt des Auftretens eines Fehlerereignisses gerade in Bearbeitung sind, an allen Speicherorten wiederholt, sobald der Fehler behoben ist.

[0040] Im Gegensatz zu den oben genannten Unzulänglichkeiten herkömmlicher Datenreplikationssysteme sind verschiedene der hier beschriebenen Ansätze in der Lage, mehr als eine Kopie von Daten an mehr als einem Datenspeicherort effizient zu verwalten. Entsprechend werden E/A-Anforderungen, die von einem Benutzer (z.B. einem Host) empfangen werden, zwischen den verschiedenen Datenspeichern implementiert, ohne dass es zu den Verzögerungen in der Durchführung kommt, die bei herkömmlichen Systemen auftreten, und ohne dass die Datenspeicherung beeinträchtigt wird, wie weiter unten noch näher beschrieben wird.

[0041] In **Fig. 4** wird ein verteiltes Datenspeichersystem 400 gemäß einem Ansatz veranschaulicht. Optional kann das vorliegende verteilte Datenspeichersystem 400 in Verbindung mit Merkmalen anderer hier aufgeführter Ansätze implementiert werden, z.B. mit denen, die unter Bezugnahme auf die anderen FIG. beschrieben werden. Ein solches verteiltes Datenspeichersystem 400 und andere hier vorgestellte Systeme können jedoch in verschiedenen Anwendungen und/oder Permutationen verwendet werden, die in den hier aufgeführten veranschaulichenden Ansätzen konkret beschrieben werden. Weiterhin kann das hier vorgestellte verteilte Datenspeichersystem 400 in jeder gewünschten Umgebung eingesetzt werden. Somit kann davon ausgegangen werden, dass **Fig. 4** (und die anderen FIG.) jede mögliche Permutation umfasst.

[0042] Wie gezeigt, umfasst das verteilte Datenspeichersystem 400 eine primäre Datenspeichereinheit 402 und eine sekundäre Datenspeichereinheit 404, die beide mit einem Netzwerk 406 verbunden sind. Ein Host 408 ist ebenfalls mit dem Netzwerk 406 verbunden, bei dem es sich um eine beliebige Art von Netzwerk handeln kann, z.B. abhängig von dem gewünschten Ansatz. In einigen Ansätzen ist das Netzwerk 406 ein WAN, z.B. das Internet. Eine beispielhafte Liste anderer Netzwerkarten, die das Netzwerk 406 implementieren kann, umfasst, ohne auf diese beschränkt zu sein, ein LAN, ein PSTN, ein SAN, ein internes Telefonnetz usw. Entsprechend sind die primäre Datenspeichereinheit 402, die sekundäre Datenspeichereinheit 404 und der Host 408 in der Lage, Daten untereinander auszutauschen, unabhängig davon, wie groß der Abstand zwischen ihnen ist, z.B. obwohl sie sich an unterschiedlichen geografischen Standorten befinden.

[0043] Jede der Datenspeichereinheiten 402, 404 und der Host 408 können in einigen Ansätzen zwar über dasselbe einzelne Netzwerk 406 Daten untereinander austauschen, es sei jedoch darauf hingewiesen, dass mehr als ein Netzwerk zwischen beliebigen zwei oder mehr der Datenspeichereinheiten 402, 404 und dem Host 408 implementiert werden kann. Beispielsweise kann der Host 408 mit jeder der Datenspeichereinheiten 402, 404 über das Netzwerk 406 Daten austauschen, während die primäre und die sekundäre Datenspeichereinheit 402, 404 über ein separates Netzwerk und/oder eine physische elektrische Verbindung, die dazwischen liegen kann, Daten untereinander austauschen. Entsprechend ist das Netzwerk 406, das in **Fig. 4** dargestellt ist, in keiner Weise als einschränkend zu verstehen und kann durchaus eine Reihe verschiedener Netzwerke umfassen, z.B. wie in **Fig. 1** dargestellt.

[0044] Sowohl die primäre als auch die sekundäre Datenspeichereinheit 402, 404 umfassen eine Steuereinheit 410 (z.B. einen Prozessor), die mit

einer Speicheranordnung 412 verbunden ist. Je nach Ansatz kann die Speicheranordnung 412, die sowohl in der primären als auch in der sekundären Datenspeichereinheit 402, 404 enthalten ist, aus verschiedenen Arten von Speicherkomponenten 414 bestehen. Beispielsweise umfasst die Speicheranordnung 412 in der primären Datenspeichereinheit 402 in einigen Ansätzen leistungsfähigere Speicherkomponenten als diejenigen, die in der sekundären Datenspeichereinheit 404 enthalten sind. Es sei darauf hingewiesen, dass „leistungsfähiger“ im Sinne der vorliegenden Beschreibung im Hinblick auf den erreichbaren Durchsatz, Verzögerungen in der Durchführung, die Zuverlässigkeitsfaktoren usw. bewertet werden kann. In anderen Worten ausgedrückt, umfasst die primäre Datenspeichereinheit 402 Speicherkomponenten, die im Vergleich zu den in der sekundären Datenspeichereinheit 404 enthaltenen Speicherkomponenten einen höheren erreichbaren Durchsatz, weniger Verzögerungen in der Durchführung, höhere Zuverlässigkeitsfaktoren usw. aufweisen.

[0045] Der Host 408 dient in einigen Ansätzen als Schnittstelle zwischen Benutzern und dem verteilten Datenspeichersystem 400. So empfängt und verarbeitet der Host 408 E/A-Anforderungen, die von einem oder mehreren Benutzern stammen. Daraus folgt, dass der Host 408 eine Steuereinheit 416 (z.B. einen Prozessor) mit einem ausreichend hohen erreichbaren Durchsatz umfasst, um die empfangenen Daten zu verarbeiten. Die Steuereinheit 416 ist weiterhin mit dem Speicher 418 verbunden, der verwendet werden kann, um Informationen (z.B. Daten, E/A-Anforderungen, Metadaten usw.) zumindest vorübergehend in einer Warteschlange zu speichern. In einigen Ansätzen kann es sich bei dem Host 408 jedoch auch um einen z14-Großrechner handeln, wie er von IBM mit einem Vertriebsbüro in 1 New Orchard Rd., Armonk, New York 10504 angeboten wird.

[0046] Wiederum sind verschiedene der hier enthaltenen Ansätze in der Lage, mehr als eine Kopie von Daten an mehr als einem Datenspeicherort effizient zu speichern. Entsprechend sind das verteilte Datenspeichersystem 400 und die darin enthaltenen Komponenten wünschenswerterweise in der Lage, E/A-Operationen zwischen primären und sekundären Datenspeichereinheiten auf effiziente und effektive Weise zu spiegeln. Sowohl die primäre als auch die sekundäre Datenspeichereinheit behalten jeweils eine Kopie derselben Daten bei und dienen dazu, die Kopien synchronisiert zu halten, selbst wenn bei einer Aktualisierung (z.B. einer Schreiboperation) Fehlerereignisse auftreten, wie an späterer Stelle noch näher beschrieben wird.

[0047] In **Fig. 5A** ist beispielsweise gemäß einem Ansatz ein Ablaufplan eines durch einen Computer

implementierten Verfahrens 500 zum Spiegeln von Schreiboperationen zwischen mehr als einer Datenspeichereinheit dargestellt. Das Verfahren 500 kann gemäß der vorliegenden Erfindung in verschiedenen Ansätzen unter anderem in jeder der in den **Fig. 1** bis **Fig. 4** abgebildeten Umgebungen durchgeführt werden. Es versteht sich, dass mehr oder weniger Operationen als die speziell in **Fig. 5A** beschriebenen im Verfahren 500 enthalten sein können, wie es für den Fachmann beim Lesen der vorliegenden Beschreibungen offensichtlich ist.

[0048] Jeder der Schritte des Verfahrens 500 kann von geeigneten Komponenten der Betriebsumgebung ausgeführt werden. So kann beispielsweise jeder der im Ablaufplan des Verfahrens 500 dargestellten Knoten 501, 502, 503 einem oder mehreren Prozessoren entsprechen, die sich in einem Datenspeichersystem mit mehreren Ebenen an einem anderen Ort befinden. Außerdem ist jeder der ein oder mehreren Prozessoren vorzugsweise so konfiguriert, dass sie miteinander Daten austauschen.

[0049] In verschiedenen Ansätzen kann das Verfahren 500 teilweise oder vollständig von einer Steuereinheit, einem Prozessor usw. oder einer anderen Einheit mit einem oder mehreren Prozessoren durchgeführt werden. Der Prozessor, z.B. Verarbeitungsschaltung(en), Chip(s) und/oder Modul(e), die in Hardware und/oder Software implementiert sind und vorzugsweise mindestens eine Hardware-Komponente aufweisen, kann in einer beliebigen Einheit verwendet werden, um einen oder mehrere Schritte des Verfahrens 500 durchzuführen. Zu veranschaulichenden Prozessoren gehören eine Zentraleinheit (CPU), eine anwendungsspezifische integrierte Schaltung (ASIC), eine vor Ort programmierbare Gatter-Anordnung (FPGA) usw., Kombinationen davon oder jede andere geeignete, nach dem Stand der Technik bekannte Datenverarbeitungseinheit, ohne auf diese beschränkt zu sein.

[0050] Wie bereits erwähnt, umfasst **Fig. 5A** verschiedene Knoten 501, 502, 503, von denen jeder einen oder mehrere Prozessoren, Steuereinheiten, Computer usw. darstellt, die sich an einem anderen Ort in einem Datenspeichersystem mit mehreren Ebenen befinden. Der Knoten 501 kann beispielsweise einen oder mehrere Prozessoren umfassen, die elektrisch mit einem Host eines verteilten Datenspeichersystems verbunden sind (siehe z.B. Host 408 in **Fig. 4** oben). Der Knoten 502 kann beispielsweise einen oder mehrere Prozessoren umfassen, die elektrisch mit einer primären Datenspeichereinheit eines verteilten Datenspeichersystems verbunden sind (siehe z.B. 402 in **Fig. 4** oben). Der Knoten 503 kann des Weiteren einen oder mehrere Prozessoren umfassen, die elektrisch mit einer zweiten Datenspeichereinheit eines verteilten Datenspeichersystems verbunden sind (siehe z.B. 404 in

Fig. 4 oben). Entsprechend können Befehle, Daten, Anforderungen usw. je nach Ansatz zwischen den einzelnen Knoten 501, 502, 503 gesendet werden. Darüber hinaus sei darauf hingewiesen, dass die verschiedenen Prozesse, die das Verfahren 500 umfasst, in keiner Weise einschränkend zu verstehen sind, wie es z.B. für den Fachmann beim Lesen der vorliegenden Beschreibung offensichtlich ist. So kann beispielsweise bei einigen Ansätzen den vom Knoten 502 an den Knoten 503 gesendeten Daten eine vom Knoten 503 an den Knoten 502 gesendete Anforderung vorangestellt werden.

[0051] Wie dargestellt, wird die Operation 504 des Verfahrens 500 von dem einen oder mehreren Prozessoren am Knoten 501 durchgeführt und umfasst Senden einer ersten Anforderung zum Durchführen einer Schreiboperation an den Knoten 502. In ähnlicher Weise wird in Operation 506 eine zweite Anforderung zum Durchführen derselben Schreiboperation an den Knoten 503 gesendet. Es sei darauf hingewiesen, dass die Anforderungen in den Operationen 504 und 506 derselben Schreiboperation entsprechen. In anderen Worten, dieselbe Schreiboperation wird sowohl an den Knoten 502 als auch an den Knoten 503 gesendet. Bei einigen Ansätzen werden je nach gewünschtem Ansatz die erste und die zweite Anforderung zum Durchführen einer Schreiboperation parallel gesendet, während bei anderen Ansätzen die erste und die zweite Anforderung zum Durchführen der Schreiboperation sequenziell, getrennt usw. gesendet werden.

[0052] Jede der Anforderungen zum Durchführen der Schreiboperation umfasst auch Metadaten, die zusammen mit den Anforderungen selbst empfangen werden. Die erste und die zweite Anforderung, die in den Operationen 504 bzw. 506 gesendet werden, entsprechen beispielsweise derselben Schreiboperation und sind daher miteinander verbunden. Diese Beziehung zwischen der ersten und der zweiten Anforderung spiegelt sich in den Metadaten wider, die zusammen mit den Anforderungen empfangen werden, und kann dazu verwendet werden, zu ermitteln, wie die beiden Anforderungen einander entsprechen.

[0053] Darüber hinaus führen die erste und die zweite Anforderung vorzugsweise dazu, dass in der primären und der sekundären Datenspeichereinheit dieselbe Schreiboperation durchgeführt wird, was als eine Form des Spiegeln, z.B. Spiegeln von Software, angesehen werden kann. Daraus folgt, dass sich der hier verwendete Begriff „Spiegeln“ auf den Prozess zum selektiven Durchführen derselben Schreiboperation in mehr als einer unterschiedlichen Datenspeichereinheit beziehen soll. Weiterhin kann (z.B. durch eine Steuereinheit) ermittelt werden, ob eine Schreiboperation „abgeglichen“ wurde, indem die Datenspeichereinheiten, an die die Schreibope-

ration gesendet wurde, abgefragt werden. Nach erfolgreichem Spiegeln der Schreiboperation können die Datenspeichereinheiten darauf reagieren, indem sie anzeigen, dass die Schreiboperation tatsächlich „abgeglichen“ wurde, z.B. wie an späterer Stelle näher beschrieben wird.

[0054] Mit Blick auf den Knoten 502 umfasst die Operation 508 als Reaktion auf Empfangen der ersten Anforderung zum Durchführen der Schreiboperation vom Host am Knoten 501 das tatsächliche Durchführen der Schreiboperation. Die Schreiboperation kann mit jedem beliebigen Datenaufzeichnungsverfahren durchgeführt werden, wie es für den Fachmann beim Lesen der vorliegenden Beschreibung offensichtlich ist. Darüber hinaus umfasst die Operation 510 Senden einer Antwort an den Host am Knoten 501, die anzeigt, dass die Schreiboperation in der primären Datenspeichereinheit erfolgreich durchgeführt wurde.

[0055] Mit Blick auf den Knoten 503 nunmehr umfasst die Operation 512 als Reaktion auf Empfangen der zweiten Anforderung zum Durchführen der Schreiboperation am Knoten 501 das tatsächliche Durchführen der Schreiboperation in der zweiten Datenspeichereinheit. Die Schreiboperation kann auch hier mit jedem beliebigen Datenaufzeichnungsverfahren durchgeführt werden, wie es für den Fachmann beim Lesen der vorliegenden Beschreibung offensichtlich ist. Darüber hinaus umfasst die Operation 514 Senden einer anderen Antwort an den Host am Knoten 501, die anzeigt, dass die Schreiboperation in der zweiten Datenspeichereinheit erfolgreich durchgeführt wurde.

[0056] Aus der Sicht des Hosts am Knoten 501 wird die Schreiboperation als tatsächlich gespiegelt ermittelt (z.B. im Kontext des Spiegeln von Software), wenn erfolgreiche Abschlussantworten (z.B. Benachrichtigungen) empfangen werden. Entsprechend wird aus der Sicht des Hosts am Knoten 501 als Reaktion auf Empfangen der Antworten in den Operationen 510, 514 festgestellt, dass die Schreiboperation erfolgreich abgeschlossen wurde. Mit Blick auf die primäre und sekundäre Speichereinheit an den Knoten 502 bzw. 503 gilt eine Schreiboperation jedoch erst dann als abgeschlossen, wenn jede Einheit von der anderen Einheit eine Anzeige empfängt, dass die Schreiboperation erfolgreich abgeschlossen wurde, wie an späterer Stelle im Detail beschrieben wird (siehe z.B. Operationen 532, 534).

[0057] Die in den Operationen 510 und 514 gesendeten Antworten können gleich, ähnlich oder voneinander verschieden sein. In anderen Worten, die Art der Antworten, die Menge der Informationen, die jede der Antworten umfasst, die Art der Informationen, die in jeder der Antworten enthalten sind, usw., die in den Operationen 510 und 514 gesendet werden, können

gleich, ähnlich oder voneinander verschieden sein. Gemäß einem beispielhaften Ansatz, der die Erfindung in keiner Weise einschränken soll, handelt es sich bei den in den Operationen 510 und 514 an den Host gesendeten Antworten um den Abschlussstatus von Kanalende/Einheitsende.

[0058] In Operation 516 wird eine Kopie der Metadaten, die der zweiten Anforderung zum Durchführen der Schreiboperation entspricht, von Knoten 503 an Knoten 502 gesendet. Die vom Knoten 503 empfangenen Metadaten werden dabei am Knoten 502 im Speicher abgelegt. Siehe Operation 518. In ähnlicher Weise wird eine Kopie der Metadaten, die der ersten Anforderung zum Durchführen der Schreiboperation entspricht, von Knoten 502 an Knoten 503 gesendet. Siehe Operation 520. In Operation 522 werden die am Knoten 503 empfangenen Metadaten weiterhin im Speicher abgelegt. Daraus folgt, dass jeder der Knoten 502, 503 Metadaten umfasst, die den beiden Anforderungen entsprechen, die der Host am Knoten 501 ausgegeben hat. Zumindest ein Teil der Metadaten, die zwischen der primären und der sekundären Datenspeichereinheit übertragen werden, ist vorzugsweise vom gleichen Typ, sodass sinnvolle Vergleiche möglich sind. Darüber hinaus können die Metadaten, die der ersten und/oder zweiten Anforderung zum Durchführen der Schreiboperation entsprechen, je nach gewünschtem Ansatz einen Zeitstempel (z.B. „Uhrzeit“-Informationen), Hashwerte oder Tabellen, die den Inhalt der Anforderungen eindeutig identifizieren, eine Speicherplatzkennung (z.B. im Format ccccCCChr) usw. umfassen.

[0059] Der Prozess zum Speichern der Metadaten im Speicher in jeder der primären und sekundären Datenspeichereinheiten kann ebenfalls je nach Ansatz variieren. Bei einigen Ansätzen werden beispielsweise Metadaten im Laufe der Zeit im Speicher angesammelt. Bei anderen Ansätzen kann die Inanspruchnahme von Datenspeicherkapazität im Vordergrund stehen, wobei nur eine bestimmte Menge an Metadaten im Speicher abgelegt sein kann. Bei wiederum anderen Ansätzen werden Metadaten, die Daten entsprechen, die infolge des Durchführens einer Schreiboperation überschrieben, gelöscht, geändert usw. wurden, von den Metadaten überschrieben, die der durchgeführten Schreiboperation entsprechen. In anderen Worten, Metadaten, die einer vorherigen Anforderung einer Schreiboperation entsprechen, werden von Metadaten überschrieben, die einer nachfolgenden Anforderung einer Schreiboperation entsprechen, die sich auf denselben Datensatz (z.B. eine Datei) bezieht wie die vorherige Anforderung. Daraus folgt, dass empfangene E/A-Anforderungen, die sich auf Metadaten beziehen, die inzwischen überschrieben wurden, bedenkenlos ignoriert werden können.

[0060] Weiterhin mit Bezug auf **Fig. 5A** kann die Kombination von Metadaten, die der ersten und zweiten Anforderung zum Durchführen der Schreiboperation entsprechen, von jedem der jeweiligen Knoten 502, 503 ausgewertet werden, um zu ermitteln, ob die Schreiboperation erfolgreich zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde. Die Entscheidung 524 wird beispielsweise von Knoten 502 getroffen und umfasst Ermitteln, ob die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit gespiegelt wurde. In anderen Worten, die Entscheidung 524 umfasst Ermitteln, ob Unterschiede zwischen den Kopien der Daten in der primären und sekundären Datenspeichereinheit abgeglichen wurden. Dieses Ermitteln erfolgt anhand der Metadaten, die in bevorzugten Ansätzen der ersten und zweiten Anforderung zum Durchführen der Schreiboperation entsprechen. So wird in einigen Ansätzen anhand von Hashwerten, die der ersten und zweiten Anforderung entsprechen, ermittelt, welche Daten in den Speicher geschrieben wurden. Darüber hinaus werden in einigen Ansätzen Metadaten, die angeben, wo die Daten in der Schreiboperation im Speicher tatsächlich abgelegt sind, zwischen der primären und sekundären Datenspeichereinheit verglichen.

[0061] In ähnlicher Weise wird die Entscheidung 526 von Knoten 503 getroffen und umfasst Ermitteln, ob die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit erfolgreich gespiegelt wurde. Wie bereits erwähnt, wird dieses Ermitteln vorzugsweise anhand der Metadaten vorgenommen, die der ersten und zweiten Anforderung zum Durchführen der Schreiboperation entsprechen, z.B. mit einem der hier beschriebenen Ansätze.

[0062] Daraus folgt, dass die Entscheidungen 524 und 526 die gleichen oder ähnliche Ermittlungen darüber umfassen, ob die Daten sowohl in der primären als auch in der sekundären Datenspeichereinheit übereinstimmen, nachdem die Schreiboperation entsprechend den ursprünglich vom Host oben empfangenen Anforderungen durchgeführt wurde. In anderen Worten, die primäre und die sekundäre Datenspeichereinheit werden miteinander synchronisiert (z.B. durch Quervergleich), um zu ermitteln, ob die Daten an beiden Orten abgeglichen wurden, wodurch sichergestellt wird, dass die Daten an beiden Orten gleich sind.

[0063] Es sei auch darauf hingewiesen, dass ausgewählte der in **Fig. 5A** dargestellten Prozesse zwar in einer bestimmten Reihenfolge oder Sequenz durchgeführt zu werden scheinen, dies jedoch in keiner Weise als Einschränkung zu verstehen ist. Mit Blick auf die Operationen 504 und 506 können die erste und die zweite Anforderung zum Durchführen derselben Schreiboperation beispielsweise gemeinsam (z.B. parallel), asynchron, sequenziell usw. gesendet

werden. In ähnlicher Weise kann die Schreiboperation asynchron in jeder der primären und sekundären Datenspeichereinheiten durchgeführt werden, z.B. je nachdem, wann die jeweilige Anforderung zum Durchführen der Schreiboperation empfangen wurde.

[0064] Wie bereits ausgeführt, wird beim Ermitteln, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde, auf Metadaten zurückgegriffen, die der an jede der Einheiten gesendeten Anforderung entsprechen. Das Durchführen der Entscheidungen 524 und/oder 526 hängt dabei in einigen Ansätzen davon ab, ob die Metadaten tatsächlich von der jeweiligen Datenspeichereinheit empfangen wurden. Bei anderen Ansätzen können die Entscheidungen 524 und/oder 526 jedoch als Reaktion auf Abwarten einer festgelegten Zeitspanne nach dem Empfangen der jeweiligen Anforderung zum Durchführen der Schreiboperation, Empfangen eines Befehls vom Host usw. getroffen werden.

[0065] Als Reaktion auf ein Feststellen, dass die Schreiboperation noch nicht zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde, geht das Verfahren 500 von Entscheidung 524 zu Operation 528. Dort umfasst die Operation 528 Abwarten einer festgelegten Zeitspanne, bevor sie zur Entscheidung 524 zurückkehrt, um zu ermitteln, ob die Unterschiede in den Daten abgeglichen wurden. Als Reaktion auf ein Feststellen, dass die Schreiboperation noch nicht zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde, geht das Verfahren 500 in ähnlicher Weise von Entscheidung 526 zu Operation 530. Dort umfasst die Operation 530 ebenfalls Abwarten einer festgelegten Zeitspanne, bevor sie zur Entscheidung 526 zurückkehrt, um zu ermitteln, ob die Unterschiede in den Daten abgeglichen wurden.

[0066] Ein nicht ordnungsgemäßes Abgleichen der Kopien von Daten in der primären und sekundären Datenspeichereinheit kann durch eine Reihe verschiedener Umstände verursacht werden. So kann beispielsweise die Verbindung zwischen dem Host und einer oder mehreren der Datenspeichereinheiten beeinträchtigt sein, sodass die Anforderung zum Durchführen der Schreiboperation bei einer oder mehreren Datenspeichereinheiten überhaupt nicht mehr empfangen wird. In solchen Situationen kann das Verfahren 500 einfach weiter darauf warten, dass die beeinträchtigte Verbindung repariert wird. In anderen Fällen kann die Verbindung zwischen der primären und der sekundären Datenspeichereinheit selbst beeinträchtigt sein, sodass ein Übertragen von Metadaten zwischen diesen Einheiten nicht möglich ist. In diesen Fällen kann das Verfahren 500 weiterhin darauf warten, dass die Verbindung zwischen den Datenspeichereinheiten selbst repa-

riert wird. In anderen Ansätzen können der Host, die primäre Datenspeichereinheit und/oder die sekundäre Datenspeichereinheit offline sein, z.B. als Reaktion auf ein Ausfallereignis. Entsprechend kann das Verfahren 500 weiterhin darauf warten, dass eine oder mehrere der physischen Komponenten selbst neu gestartet und/oder repariert werden. Als Reaktion auf Rückkehren in den Normalbetrieb können Schreiboperationen, die auf einer funktionierenden Einheit implementiert sind, während die ausgefallene Einheit offline war, speziell gekennzeichnet (z.B. in einem Delta-Aktualisierungsdokument) und in einem inkrementellen Neusynchronisationsverfahren implementiert werden.

[0067] In Situationen, in denen die Operation 528 und/oder 530 eine festgelegte Anzahl von Malen wiederholt wird/werden, kann das Verfahren 500 zusätzliche Maßnahmen ergreifen. Wenn beispielsweise festgestellt wird, dass die Daten in der primären und sekundären Datenspeichereinheit nach einer festgelegten Anzahl von Wiederholungsversuchen immer noch nicht synchronisiert sind, kann die Spiegelungsbeziehung zwischen der primären und sekundären Datenspeichereinheit beendet werden, z.B. zumindest vorübergehend, sodass ein Datenwiederherstellungsprotokoll eingegeben werden kann. Der Grund dafür, dass das Datenwiederherstellungsprotokoll implementiert wird, besteht darin, dass die Schreiboperation nicht erfolgreich zwischen den Datenspeichereinheiten gespiegelt wurde. Das Datenwiederherstellungsprotokoll kann zum Beispiel Überprüfen und Reparieren der Verbindung zwischen dem Host und den Datenspeichereinheiten umfassen. Bei anderen Ansätzen kann ein erneuter Versuch unternommen werden, Metadaten zu übertragen, die den Anforderungen der Schreiboperationen zwischen der primären und sekundären Datenspeichereinheit entsprechen. Bei anderen Ansätzen kann der Host jedoch aufgefordert werden, die Anforderungen für die Schreiboperationen insgesamt neu auszugeben.

[0068] In anderen Ansätzen kann das Wiederherstellungsprotokoll sicherstellen, dass die der Schreiboperation entsprechenden Daten erfolgreich zwischen der primären und sekundären Datenspeichereinheit gespiegelt werden, wodurch das Verfahren 500 beendet werden kann. Es sei jedoch darauf hingewiesen, dass das Verfahren 500 nach Durchführen einer bestimmten Operation zwar tatsächlich beendet sein kann, ein oder mehrere der in Verfahren 500 enthaltenen Prozesse jedoch wiederholt werden können, um zusätzliche Schreiboperationen zwischen den Datenspeichereinheiten zu spiegeln. In anderen Worten, ein oder mehrere der im Verfahren 500 enthaltenen Prozesse können für nachfolgend empfangene Schreiboperationen wiederholt werden.

[0069] Zurück zu Entscheidung 526: Als Reaktion auf ein Feststellen, dass die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde, geht das Verfahren zu Operation 532. Dort umfasst die Operation 532 Senden einer Antwort an die primäre Datenspeichereinheit am Knoten 502, die anzeigt, dass die sekundäre Datenspeichereinheit festgestellt hat, dass die Schreiboperation erfolgreich zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurde. In anderen Worten, eine Schreibabschlussbestätigung wird von der sekundären Datenspeichereinheit an die primäre Datenspeichereinheit gesendet, die anzeigt, dass die Daten zwischen den Datenspeichereinheiten in Übereinstimmung mit der sekundären Datenspeichereinheit abgeglichen wurden. Entsprechend wird die primäre Datenspeichereinheit darüber informiert, dass die sowohl in der primären als auch in der sekundären Datenspeichereinheit gespeicherten Daten in Übereinstimmung mit der Schreiboperation aktualisiert wurden. Als Reaktion auf ein Feststellen, dass die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde, geht das Verfahren 500 in ähnlicher Weise von Entscheidung 534 zu Operation 524. Dort umfasst die Operation 534 Senden einer anderen Antwort an die primäre Datenspeichereinheit am Knoten 503, die anzeigt, dass die primäre Datenspeichereinheit festgestellt hat, dass die Schreiboperation erfolgreich zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurde.

[0070] Wenn die primäre und die sekundäre Datenspeichereinheit Antworten voneinander empfangen, die anzeigen, dass die Schreiboperation erfolgreich zwischen ihnen abgeglichen wurde, kann daraus geschlossen werden, dass die Daten in jeder der Datenspeichereinheiten erfolgreich abgeglichen wurden. Entsprechend können auf der Grundlage dieser Informationen weitere Schritte wie z.B. Senden eines nachfolgenden Schreibbefehls, Verbessern eines E/A-Puffers, Aktualisieren einer logisch-physischen Tabelle usw. wie gewünscht durchgeführt werden.

[0071] Daraus folgt, dass die verschiedenen oben beschriebenen Ansätze in Bezug auf das Verfahren 500 wünschenswerterweise in der Lage sind, die Durchführungszeiten zu verkürzen, indem sie es dem Host ermöglichen, mit der primären und sekundären Datenspeichereinheit parallel Daten auszutauschen. Verschiedene der hier beschriebenen Ansätze sind beispielsweise in der Lage, den Gesamtaufwand im Vergleich zu herkömmlichen Datenreplikationssystemen zu verringern. Ebenso können die primäre und die sekundäre Datenspeichereinheit parallel arbeiten und zwischen diesen Synchronisationsvorgänge durchführen, ohne auf Eingaben des Hosts angewiesen zu sein. Auf diese

Weise können die Datenspeichereinheiten dazu dienen, dass die Daten entsprechend den vorgesehenen Datenspeicherschemata weitergegeben werden. Dadurch werden der Netzwerkverkehr verringert und die Datenspeicherung verbessert, da aktiv sichergestellt wird, dass Daten zwischen mehreren Datenspeichereinheiten gespiegelt werden. Entsprechend ist es vorzuziehen, dass die verschiedenen Prozesse in diesen Ansätzen auf alle Schreiboperationen (z.B. alle Anwendungsschreiboperationen) angewendet werden, die vom Host empfangen werden.

[0072] Wie bereits erwähnt, können während des Prozesses zum Spiegeln einer Schreiboperation zwischen der primären und sekundären Datenspeichereinheit Fehler auftreten. In diesem Fall können verschiedene Datenwiederherstellungsprotokolle implementiert werden, um die Datenübertragungskanäle zwischen den Orten im übergeordneten verteilten Speichersystem, einem zentralen Netzwerk, fehlerhaften physischen Komponenten und/oder darauf implementierter Software usw. zu reparieren. Außerdem können weitere Anforderungen zum Durchführen von Schreiboperationen empfangen werden, bevor eine aktuelle Schreiboperation erfolgreich implementiert wurde. Diese Anforderungen für Schreiboperationen werden vorzugsweise so verarbeitet und gestaffelt, dass die entsprechenden Metadaten im Speicher abgelegt und für die Implementierung eingeplant werden. Darüber hinaus werden Anforderungen für Leseoperationen, die empfangen werden, während eine Schreiboperation durchgeführt wird, ebenfalls vorzugsweise zurückgestellt und zu einem späteren Zeitpunkt durchgeführt, wie z.B. aus der nachfolgenden Beschreibung ersichtlich wird.

[0073] **Fig. 5B** veranschaulicht beispielsweise ein Verfahren 550 zum Auswerten einer empfangenen Leseanforderung gemäß einem Ansatz. Das Verfahren 550 wird im Folgenden im Zusammenhang mit dem verteilten Datenspeichersystem vorgestellt, auf das oben in Verfahren 500 Bezug genommen wird, und umfasst somit die verschiedenen in **Fig. 4** dargestellten Komponenten. Jeder der in Verfahren 550 enthaltenen Prozesse kann gemäß der vorliegenden Erfindung in verschiedenen Ansätzen unter anderem in jeder der in den **Fig. 1** bis **Fig. 5A** abgebildeten Umgebungen durchgeführt werden. Es versteht sich, dass mehr oder weniger Operationen als die speziell in **Fig. 5B** beschriebenen im Verfahren 550 enthalten sein können, wie es für den Fachmann beim Lesen der vorliegenden Beschreibungen offensichtlich ist.

[0074] Jeder der Schritte des Verfahrens 550 kann von geeigneten Komponenten der Betriebsumgebung ausgeführt werden. In verschiedenen Ansätzen kann das Verfahren 550 beispielsweise teilweise

oder vollständig von einer Steuereinheit, einem Prozessor usw. oder einer anderen Einheit mit einem oder mehreren Prozessoren durchgeführt werden. Der Prozessor, z.B. Verarbeitungsschaltung(en), Chip(s) und/oder Modul(e), die in Hardware und/oder Software implementiert sind und vorzugsweise mindestens eine Hardware-Komponente aufweisen, kann in einer beliebigen Einheit verwendet werden, um einen oder mehrere Schritte des Verfahrens 550 durchzuführen. Zu veranschaulichenden Prozessoren gehören eine Zentraleinheit (CPU), eine anwendungsspezifische integrierte Schaltung (ASIC), eine vor Ort programmierbare Gatter-Anordnung (FPGA) usw., Kombinationen davon oder jede andere geeignete, nach dem Stand der Technik bekannte Datenverarbeitungseinheit, ohne auf diese beschränkt zu sein.

[0075] Wie in **Fig. 5B** dargestellt, umfasst die Operation 552 des Verfahrens 550 Empfangen einer Leseanforderung. Die Leseanforderung kann je nach Ansatz in der primären und/oder sekundären Datenspeichereinheit empfangen werden. Die primäre Datenspeichereinheit kann z.B. als aktive Datenspeichereinheit dienen, die für Antworten auf Datenanfragen, z.B. Leseanforderungen, die während des Systembetriebs empfangen werden, zuständig ist. Außerdem kann zu jedem beliebigen Zeitpunkt eine Leseanforderung vom Host empfangen werden. Eine Leseanforderung für eine bestimmte Datensammlung (z.B. eine Datei oder einen Datensatz) kann sogar empfangen werden, bevor diese Daten zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurden.

[0076] Entsprechend umfasst die Entscheidung 554 Ermitteln, ob die in der Leseanforderung angegebenen Daten zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden. In anderen Worten, die Entscheidung 554 umfasst Ermitteln, ob die in der Leseanforderung angegebenen Daten bereits sowohl in die primäre als auch in die sekundäre Datenspeichereinheit geschrieben wurden, und somit, ob die Daten aus Sicht der Speichereinheit abgeglichen wurden, bevor zugelassen wird, dass die Leseanforderung erfüllt wird. Bei einigen Ansätzen erfolgt dieses Ermitteln durch Einsichtnahme in eine logisch-physische Tabelle, die dazu dient, zu erfassen, welche Daten bereits zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden. Das Ermitteln kann auch in anderen Ansätzen erfolgen, indem ein E/A-Protokoll eingesehen wird, das die in der Speichereinheit empfangenen Anforderungen für Schreiboperationen enthält, wobei der Speicher konkret untersucht wird (z.B. indem Testleseoperationen durchgeführt werden) usw. Es sei auch darauf hingewiesen, dass die Spiegelungsbeziehung zwischen der primären und der sekundären Datenspeichereinheit weiterhin besteht,

sodass ein erfolgreiches Abgleichen der Daten in jeder der Datenspeichereinheiten gewährleistet ist.

[0077] Als Reaktion auf ein Feststellen, dass die in der Leseanforderung angegebenen Daten noch nicht zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden, geht das Verfahren 550 von Entscheidung 554 zu Operation 556. Dort umfasst die Operation 556 Verzögern der Leseanforderung. Die Leseanforderung wird vorzugsweise verzögert, bis die in der Leseanforderung angegebene Daten zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden. Entsprechend kann die Leseanforderung vorübergehend für eine festgelegte Zeit in einem Puffer gehalten werden.

[0078] Von der Operation 556 kehrt das Verfahren 550 zur Entscheidung 554 zurück, sodass das Ermitteln für die der Leseanforderung entsprechenden Daten wiederholt werden kann. Daraus folgt, dass die Prozesse 554 und 556 beliebig oft wiederholt werden können, bis festgestellt wird, dass die in der Leseanforderung angegebenen Daten zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden. Bei einigen Ansätzen kann jedoch eine Antwort, die anzeigt, dass die Leseanforderung fehlgeschlagen ist, an den Host zurückgesendet werden, nachdem die Entscheidung 554 eine vorgegebene Anzahl von Malen durchgeführt wurde, nachdem eine vorgegebene Zeitspanne verstrichen ist, auf die angeforderten Daten nicht zugegriffen werden kann (z.B. weil ein Speicherfehler und/oder ein Datenverlust ermittelt wurde), usw.

[0079] Das Verfahren 550 geht jedoch von der Entscheidung 554 zur Operation 558 weiter, wenn festgestellt wird, dass die in der Leseanforderung angegebenen Daten zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurden. Dort umfasst die Operation 558 konkret Durchführen der Leseoperation durch Zugreifen auf die angeforderten Daten aus dem Speicher und Bereitstellen (z.B. Zurücksenden) der angeforderten Daten an den Host. Wie bereits ausgeführt, dient entweder die primäre oder die sekundäre Datenspeichereinheit als „aktive“ Einheit, mit der empfangene Leseanforderungen durchgeführt werden, während die jeweils andere Datenspeichereinheit als „redundante“ Einheit dient. Entsprechend können die Prozesse des Verfahrens 550 von der jeweils aktiven Datenspeichereinheit durchgeführt werden. Wenn jedoch die aktive Datenspeichereinheit offline geht, z.B. nach einem Ausfallereignis, kann die redundante Einheit die Zuständigkeit für den Betrieb übernehmen und zumindest vorübergehend die aktive Einheit werden. In ähnlicher Weise werden dauerhafte Fehler, die auf der aktiven Einheit auftreten, vorzugsweise zum Implementieren an die redundante Einheit weitergeleitet.

[0080] Daraus folgt, dass die Prozesse in Verfahren 550 die Betriebseffizienz des übergeordneten Systems ermöglichen, indem sie im Wesentlichen Situationen ausschließen, in denen eine Leseoperation versucht wird, nur um festzustellen, dass die Daten, die gelesen werden sollen, noch nicht im Speicher gespeichert sind, noch nicht in Übereinstimmung mit einer modifizierten Schreiboperation aktualisiert wurden usw. Darüber hinaus kann das Verfahren 550 in Verbindung mit dem Verfahren 500 implementiert werden, ohne dass das Durchführen unterbrochen wird. Dies ist besonders wünschenswert, da die Verringerungen der Verzögerung und die Datenspeicherung, die durch die oben beschriebenen Ansätze in Bezug auf das Verfahren 500 erreicht werden, mit den Verbesserungen bei der Durchführung der Leseoperation kombiniert werden, die durch die in Bezug auf das Verfahren 550 beschriebenen Ansätze erreicht werden.

[0081] Wiederum sind verschiedene der hier beschriebenen Ansätze wünschenswerterweise in der Lage, die Durchführungszeiten zu verkürzen, da der Host parallel mit den mehreren Datenspeichereinheiten Daten austauschen kann. Ebenso können die Datenspeichereinheiten parallel arbeiten und zwischen diesen Synchronisationsvorgänge durchführen, ohne auf Eingaben des Hosts angewiesen zu sein. Auf diese Weise können die Steuereinheiten in jeder Datenspeichereinheit dazu dienen, dass die Daten entsprechend den vorgesehenen Datenspeicherschemata weitergegeben werden. Außerdem sind die Steuereinheiten in der Lage, asynchron Daten auszutauschen, wenn E/A-Anforderungen implementiert werden. Dadurch werden der Netzwerkverkehr verringert und die Datenspeicherung verbessert, da aktiv sichergestellt wird, dass Daten zwischen mehreren Datenspeichereinheiten gespiegelt werden. Entsprechend ist es vorzuziehen, dass die verschiedenen Prozesse in diesen Ansätzen auf alle Schreiboperationen (z.B. alle Anwendungsschreiboperationen) angewendet werden, die vom Host empfangen werden.

[0082] Es sei auch darauf hingewiesen, dass die verschiedenen hier beschriebenen Ansätze in keiner Weise dazu dienen, die Erfindung einzuschränken. Viele der oben beschriebenen Ansätze werden zwar im Zusammenhang mit einem verteilten Datenspeichersystem mit einem Host und zwei Datenspeichereinheiten beschrieben, doch kann beispielsweise eine beliebige Anzahl von Datenspeichereinheiten implementiert werden. So kann beispielsweise eine dritte Kopie bestimmter Daten (z.B. Daten, die als besonders „wichtig“ eingestuft werden) in einer dritten Datenspeichereinheit gespeichert werden, die mit demselben Netzwerk verbunden ist wie die anderen Speichereinheiten. Auch wenn die oben beschriebenen Prozesse für jede empfangene E/A-Anforderung durchgeführt werden,

kann eine beliebige Anzahl von E/A-Anforderungen zusammengefasst und gemeinsam verarbeitet werden.

[0083] Bei der vorliegenden Erfindung kann es sich um ein System, ein Verfahren und/oder ein Computerprogrammprodukt auf jedem möglichen technischen Detaillierungsgrad der Integration handeln. Das Computerprogrammprodukt kann (ein) durch einen Computer lesbare(s) Speichermedium (oder -medien) umfassen, auf dem/denen durch einen Computer lesbare Programmanweisungen gespeichert ist/sind, um einen Prozessor dazu zu veranlassen, Aspekte der vorliegenden Erfindung auszuführen.

[0084] Bei dem durch einen Computer lesbaren Speichermedium kann es sich um eine physische Einheit handeln, die Anweisungen zur Verwendung durch eine Einheit zur Ausführung von Anweisungen behalten und speichern kann. Bei dem durch einen Computer lesbaren Speichermedium kann es sich zum Beispiel um eine elektronische Speichereinheit, eine magnetische Speichereinheit, eine optische Speichereinheit, eine elektromagnetische Speichereinheit, eine Halbleiterspeichereinheit oder jede geeignete Kombination daraus handeln, ohne auf diese beschränkt zu sein. Zu einer nicht erschöpfenden Liste spezifischerer Beispiele des durch einen Computer lesbaren Speichermediums gehören die Folgenden: eine tragbare Computerdiskette, eine Festplatte, ein Direktzugriffsspeicher (RAM), ein Nur-Lese-Speicher (ROM), ein löschbarer programmierbarer Nur-Lese-Speicher (EPROM bzw. Flash-Speicher), ein statischer Direktzugriffsspeicher (SRAM), ein tragbarer Kompaktspeicherplatte-Nur-Lese-Speicher (CD-ROM), eine DVD (digital versatile disc), ein Speicher-Stick, eine Diskette, eine mechanisch codierte Einheit wie zum Beispiel Lochkarten oder gehobene Strukturen in einer Rille, auf denen Anweisungen gespeichert sind, und jede geeignete Kombination daraus. Ein durch einen Computer lesbares Speichermedium soll in der Verwendung hier nicht als flüchtige Signale an sich aufgefasst werden, wie zum Beispiel Funkwellen oder andere sich frei ausbreitende elektromagnetische Wellen, elektromagnetische Wellen, die sich durch einen Wellenleiter oder ein anderes Übertragungsmedium ausbreiten (z.B. einen Lichtwellenleiter durchlaufende Lichtimpulse) oder durch einen Draht übertragene elektrische Signale.

[0085] Hier beschriebene, durch einen Computer lesbare Programmanweisungen können von einem durch einen Computer lesbaren Speichermedium auf jeweilige Datenverarbeitungs/Verarbeitungs-Einheiten oder über ein Netzwerk wie zum Beispiel das Internet, ein lokales Netzwerk, ein Weitverkehrsnetzwerk und/oder ein drahtloses Netzwerk auf einen externen Computer oder eine externe Speicherein-

heit heruntergeladen werden. Das Netzwerk kann Kupferübertragungskabel, Lichtwellenübertragungsleiter, drahtlose Übertragung, Router, Firewalls, Vermittlungseinheiten, Gateway-Computer und/oder Edge-Server aufweisen. Eine Netzwerkkartenskarte oder Netzwerkschnittstelle in jeder Datenverarbeitungs/Verarbeitungs-Einheit empfängt durch einen Computer lesbare Programmanweisungen aus dem Netzwerk und leitet die durch einen Computer lesbaren Programmanweisungen zur Speicherung in einem durch einen Computer lesbaren Speichermedium innerhalb der entsprechenden Datenverarbeitungs/Verarbeitungs-Einheit weiter.

[0086] Bei durch einen Computer lesbaren Programmanweisungen zum Ausführen von Arbeitsschritten der vorliegenden Erfindung kann es sich um Assembler-Anweisungen, ISA-Anweisungen (Instruction-Set-Architecture), Maschinenanweisungen, maschinenabhängige Anweisungen, Mikrocode, Firmware-Anweisungen, zustandssetzende Daten, Konfigurationsdaten für integrierte Schaltungen oder entweder Quellcode oder Objektcode handeln, die in einer beliebigen Kombination aus einer oder mehreren Programmiersprachen geschrieben werden, darunter objektorientierte Programmiersprachen wie Smalltalk, C++ o. ä. sowie prozedurale Programmiersprachen wie die Programmiersprache „C“ oder ähnliche Programmiersprachen. Die durch einen Computer lesbaren Programmanweisungen können vollständig auf dem Computer des Benutzers, teilweise auf dem Computer des Benutzers, als eigenständiges Software-Paket, teilweise auf dem Computer des Benutzers und teilweise auf einem entfernt angeordneten Computer oder vollständig auf dem entfernt angeordneten Computer oder Server ausgeführt werden. In letzterem Fall kann der entfernt angeordnete Computer mit dem Computer des Benutzers durch eine beliebige Art Netzwerk verbunden sein, darunter ein lokales Netzwerk (LAN) oder ein Weitverkehrsnetzwerk (WAN), oder die Verbindung kann mit einem externen Computer hergestellt werden (zum Beispiel über das Internet unter Verwendung eines Internet-Dienstansbieters). In einigen Ausführungsformen können elektronische Schaltungen, darunter zum Beispiel programmierbare Logikschaltungen, vor Ort programmierbare Gatter-Anordnungen (FPGA, field programmable gate arrays) oder programmierbare Logikanordnungen (PLA, programmable logic arrays) die durch einen Computer lesbaren Programmanweisungen ausführen, indem sie Zustandsinformationen der durch einen Computer lesbaren Programmanweisungen nutzen, um die elektronischen Schaltungen zu personalisieren, um Aspekte der vorliegenden Erfindung durchzuführen.

[0087] Aspekte der vorliegenden Erfindung sind hier unter Bezugnahme auf Ablaufpläne und/oder Blockschaltbilder bzw. Schaubilder von Verfahren, Vor-

richtungen (Systemen) und Computerprogrammprodukten gemäß Ausführungsformen der Erfindung beschrieben. Es wird darauf hingewiesen, dass jeder Block der Ablaufpläne und/oder der Blockschaltbilder bzw. Schaubilder sowie Kombinationen von Blöcken in den Ablaufplänen und/oder den Blockschaltbildern bzw. Schaubildern mittels durch einen Computer lesbare Programmanweisungen ausgeführt werden können.

[0088] Diese durch einen Computer lesbaren Programmanweisungen können einem Prozessor eines Computers oder einer anderen programmierbaren Datenverarbeitungsvorrichtung bereitgestellt werden, um eine Maschine zu erzeugen, sodass die über den Prozessor des Computers bzw. der anderen programmierbaren Datenverarbeitungsvorrichtung ausgeführten Anweisungen ein Mittel zur Umsetzung der in dem Block bzw. den Blöcken der Ablaufpläne und/oder der Blockschaltbilder bzw. Schaubilder festgelegten Funktionen/Schritte erzeugen. Diese durch einen Computer lesbaren Programmanweisungen können auch auf einem durch einen Computer lesbaren Speichermedium gespeichert sein, das einen Computer, eine programmierbare Datenverarbeitungsvorrichtung und/oder andere Einheiten so steuern kann, dass sie auf eine bestimmte Art funktionieren, sodass das durch einen Computer lesbare Speichermedium, auf dem Anweisungen gespeichert sind, einen Herstellungsartikel aufweist, darunter Anweisungen, welche Aspekte der/des in dem Block bzw. den Blöcken des Ablaufplans und/oder der Blockschaltbilder bzw. Schaubilder angegebenen Funktion/Schritts umsetzen.

[0089] Die durch einen Computer lesbaren Programmanweisungen können auch auf einen Computer, eine andere programmierbare Datenverarbeitungsvorrichtung oder eine andere Einheit geladen werden, um das Ausführen einer Reihe von Prozessschritten auf dem Computer bzw. der anderen programmierbaren Vorrichtung oder anderen Einheit zu verursachen, um einen auf einem Computer ausgeführten Prozess zu erzeugen, sodass die auf dem Computer, einer anderen programmierbaren Vorrichtung oder einer anderen Einheit ausgeführten Anweisungen die in dem Block bzw. den Blöcken der Ablaufpläne und/oder der Blockschaltbilder bzw. Schaubilder festgelegten Funktionen/Schritte umsetzen.

[0090] Die Ablaufpläne und die Blockschaltbilder bzw. Schaubilder in den Figuren veranschaulichen die Architektur, die Funktionalität und den Betrieb möglicher Ausführungen von Systemen, Verfahren und Computerprogrammprodukten gemäß verschiedener Ausführungsformen der vorliegenden Erfindung. In diesem Zusammenhang kann jeder Block in den Ablaufplänen oder Blockschaltbildern bzw.

Schaubildern ein Modul, ein Segment oder einen Teil von Anweisungen darstellen, die eine oder mehrere ausführbare Anweisungen zur Ausführung der bestimmten logischen Funktion(en) aufweisen. In einigen alternativen Ausführungen können die in dem Block angegebenen Funktionen in einer anderen Reihenfolge als in den Figuren gezeigt stattfinden. Zwei nacheinander gezeigte Blöcke können zum Beispiel in Wirklichkeit als ein Schritt ausgeführt, gleichzeitig ausgeführt, im Wesentlichen gleichzeitig ausgeführt, ganz oder teilweise zeitlich überlappend ausgeführt werden, oder die Blöcke können manchmal je nach entsprechender Funktionalität in umgekehrter Reihenfolge ausgeführt werden. Es ist ferner anzumerken, dass jeder Block der Blockschaltbilder bzw. Schaubilder und/oder der Ablaufpläne sowie Kombinationen aus Blöcken in den Blockschaltbildern bzw. Schaubildern und/oder den Ablaufplänen durch spezielle auf Hardware beruhende Systeme umgesetzt werden können, welche die festgelegten Funktionen oder Schritte durchführen, oder Kombinationen aus Spezial-Hardware und Computeranweisungen ausführen.

[0091] Darüber hinaus kann ein System gemäß verschiedenen Ausführungsformen einen Prozessor und eine in den Prozessor integrierte und/oder von ihm ausführbare Logik enthalten, wobei die Logik so konfiguriert ist, dass sie einen oder mehrere der hier beschriebenen Prozessschritte ausführt. Der Prozessor kann jede hier beschriebene Konfiguration aufweisen, z.B. einen diskreten Prozessor oder einen Verarbeitungsschaltkreis, der viele Komponenten umfasst, z.B. Verarbeitungshardware, Speicher, E/A-Schnittstellen usw. Mit „integriert in“ ist gemeint, dass der Prozessor eine eingebettete Logik als Hardware-Logik hat, wie z.B. eine anwendungsspezifische integrierte Schaltung (ASIC), eine FPGA usw. Mit „ausführbar durch den Prozessor“ ist gemeint, dass die Logik Hardware-Logik ist; Software-Logik wie z.B. Firmware, Teil eines Betriebssystems, Teil eines Anwendungsprogramms; usw. oder eine Kombination aus Hardware- und Software-Logik, auf die der Prozessor zugreifen kann und die so konfiguriert ist, dass der Prozessor veranlasst wird, beim Ausführen durch den Prozessor eine bestimmte Funktionalität auszuführen. Software-Logik kann auf einem lokalen und/oder entfernt angeordneten Speicher eines beliebigen Speichertyps gespeichert werden, wie er nach dem Stand der Technik bekannt ist. Jeder nach dem Stand der Technik bekannte Prozessor kann verwendet werden, z.B. ein Software-Prozessormodul und/oder ein Hardware-Prozessor wie eine ASIC, eine FPGA, eine Zentraleinheit (CPU), eine integrierte Schaltung (integrated circuit, IC), eine Grafikverarbeitungseinheit (graphics processing unit, GPU) usw.

[0092] Es ist offensichtlich, dass die verschiedenen Merkmale der oben genannten Systeme und/oder

Methodiken in beliebiger Weise kombiniert werden können, sodass sich aus den obigen Beschreibungen eine Mehrzahl an Kombinationen ergibt.

[0093] Es ist weiterhin offensichtlich, dass Ausführungsformen der vorliegenden Erfindung in Form eines Dienstes zur Verfügung gestellt werden können, der für einen Kunden eingesetzt wird, um eine Dienstleistung auf Anfrage anzubieten.

[0094] Die Beschreibungen der verschiedenen Ausführungsformen der vorliegenden Erfindung wurden zum Zwecke der Veranschaulichung vorgestellt, sollen jedoch nicht erschöpfend oder auf die Ausführungsformen beschränkt sein. Für Fachleute ist offensichtlich, dass viele Änderungen und Abwandlungen möglich sind, ohne vom Anwendungsbereich der beschriebenen Ausführungsformen abzuweichen. Die hierin verwendete Terminologie wurde gewählt, um die Grundgedanken der Ausführungsformen, die praktische Anwendung oder technische Verbesserung gegenüber Technologien auf dem Markt bestmöglich zu erläutern oder es Fachleuten zu ermöglichen, die hierin beschriebenen Ausführungsformen zu verstehen.

Patentansprüche

1. Durch einen Computer implementiertes Verfahren zum Spiegeln von Schreiboperationen zwischen primärer und sekundärer Datenspeichereinheit, das aufweist:
Empfangen einer ersten Anforderung zum Durchführen einer Schreiboperation von einem Host;
Durchführen der Schreiboperation;
Senden von Metadaten, die der ersten Anforderung entsprechen, an die sekundäre Datenspeichereinheit;
Empfangen von Metadaten, die einer zweiten Anforderung zum Durchführen der Schreiboperation entsprechen, von der sekundären Datenspeichereinheit, wobei die zweite Anforderung vom Host in der sekundären Datenspeichereinheit empfangen wurde;
Verwenden der Metadaten, die der ersten und zweiten Anforderung entsprechen, um zu ermitteln, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde; und
als Reaktion auf ein Feststellen, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit gespiegelt wurde, Senden einer Antwort an die sekundäre Datenspeichereinheit, die anzeigt, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurde.

2. Durch einen Computer implementiertes Verfahren nach Anspruch 1, das aufweist:
Empfangen einer Antwort von der sekundären

Datenspeichereinheit, die anzeigt, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurde, als Reaktion auf ein Feststellen durch die sekundäre Datenspeichereinheit, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit gespiegelt wurde, wobei die sekundäre Datenspeichereinheit die Schreiboperation als Reaktion auf Empfangen der zweiten Anforderung zum Durchführen der Schreiboperation durchführt, wobei die sekundäre Datenspeichereinheit die Metadaten verwendet, die der ersten und zweiten Anforderung entsprechen, um zu ermitteln, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde.

3. Durch einen Computer implementiertes Verfahren nach Anspruch 1, das aufweist:
Empfangen einer Leseanforderung;
Ermitteln, ob in der Leseanforderung angegebene Daten zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden;
als Reaktion auf ein Feststellen, dass die in der Leseanforderung angegebenen Daten nicht zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden, Verzögern des Durchführens der Leseanforderung; und
als Reaktion auf ein Feststellen, dass die in der Leseanforderung angegebenen Daten zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden, Durchführen der Leseoperation.

4. Durch einen Computer implementiertes Verfahren nach Anspruch 1, das aufweist:
Senden einer Antwort an den Host, die anzeigt, dass die Schreiboperation durchgeführt wurde, bevor ermittelt wurde, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurde.

5. Durch einen Computer implementiertes Verfahren nach Anspruch 1, wobei die Metadaten, die der ersten Anforderung entsprechen, Informationen umfassen, die aus der Gruppe ausgewählt wurden, die besteht aus: einem Zeitstempel, Hashwerten und einer Speicherortkennung, wobei die Metadaten, die der zweiten Anforderung entsprechen, Informationen umfassen, die aus der Gruppe ausgewählt wurden, die besteht aus: einem Zeitstempel, Hashwerten und einer Speicherortkennung.

6. Durch einen Computer implementiertes Verfahren nach Anspruch 1, das aufweist:
Speichern der Metadaten, die der ersten Anforderung entsprechen, im Speicher, wobei die Metadaten, die der ersten Anforderung entsprechen, von Metadaten überschrieben werden, die einer nachfolgenden Anforderung entsprechen,

eine Schreiboperation in demselben Datensatz wie die erste Anforderung durchzuführen.

7. Durch einen Computer implementiertes Verfahren nach Anspruch 1, wobei die Prozesse für alle vom Host empfangenen Anforderungen für Schreiboperationen durchgeführt werden.

8. Computerprogrammprodukt zum Spiegeln von Schreiboperationen zwischen primärer und sekundärer Datenspeichereinheit, wobei das Computerprogrammprodukt ein durch einen Computer lesbares Speichermedium mit darauf enthaltenen Programmanweisungen aufweist, wobei die Programmanweisungen von einem Prozessor lesbar und/oder ausführbar sind, um den Prozessor zu veranlassen:

eine erste Anforderung zum Durchführen einer Schreiboperation von einem Host durch den Prozessor zu empfangen;

die Schreiboperation durch den Prozessor durchzuführen;

Metadaten, die der ersten Anforderung entsprechen, durch den Prozessor an die sekundäre Datenspeichereinheit zu senden;

Metadaten, die einer zweiten Anforderung zum Durchführen der Schreiboperation entsprechen, von der sekundären Datenspeichereinheit durch den Prozessor zu empfangen, wobei die zweite Anforderung vom Host in der sekundären Datenspeichereinheit empfangen wurde;

die Metadaten, die der ersten und zweiten Anforderung entsprechen, durch den Prozessor zu verwenden, um zu ermitteln, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde; und

als Reaktion auf ein Feststellen, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit gespiegelt wurde, eine Antwort an die sekundäre Datenspeichereinheit durch den Prozessor zu senden, die anzeigt, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurde.

9. Computerprogrammprodukt nach Anspruch 8, wobei die Programmanweisungen von einem Prozessor lesbar und/oder ausführbar sind, um den Prozessor zu veranlassen:

eine Antwort von der sekundären Datenspeichereinheit durch den Prozessor zu empfangen, die anzeigt, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurde, als Reaktion auf ein Feststellen durch die sekundäre Datenspeichereinheit, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit gespiegelt wurde, wobei die sekundäre Datenspeichereinheit die Schreiboperation als Reaktion auf Empfangen der zweiten Anforderung zum Durchführen der Schreib-

operation durchführt,

wobei die sekundäre Datenspeichereinheit die Metadaten verwendet, die der ersten und zweiten Anforderung entsprechen, um zu ermitteln, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde.

10. Computerprogrammprodukt nach Anspruch 8, wobei die Programmanweisungen von dem Prozessor lesbar und/oder ausführbar sind, um den Prozessor zu veranlassen:

eine Leseanforderungen durch den Prozessor zu empfangen;

durch den Prozessor zu ermitteln, ob in der Leseanforderung angegebene Daten zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden;

als Reaktion auf ein Feststellen, dass die in der Leseanforderung angegebenen Daten nicht zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden, ein Durchführen der Leseanforderung durch den Prozessor zu verzögern; und

als Reaktion auf ein Feststellen, dass die in der Leseanforderung angegebenen Daten zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden, die Leseoperation durch den Prozessor durchzuführen.

11. Computerprogrammprodukt nach Anspruch 8, wobei die Programmanweisungen von dem Prozessor lesbar und/oder ausführbar sind, um den Prozessor zu veranlassen:

eine Antwort an den Host zu senden, die anzeigt, dass die Schreiboperation durchgeführt wurde, bevor ermittelt wurde, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurde.

12. Computerprogrammprodukt nach Anspruch 8, wobei die Metadaten, die der ersten Anforderung entsprechen, Informationen umfassen, die aus der Gruppe ausgewählt wurden, die besteht aus: einem Zeitstempel, Hashwerten und einer Speicherortkennung, wobei die Metadaten, die der zweiten Anforderung entsprechen, Informationen umfassen, die aus der Gruppe ausgewählt wurden, die besteht aus: einem Zeitstempel, Hashwerten und einer Speicherortkennung.

13. Computerprogrammprodukt nach Anspruch 8, wobei die Programmanweisungen von dem Prozessor lesbar und/oder ausführbar sind, um den Prozessor zu veranlassen:

die Metadaten, die der ersten Anforderung entsprechen, durch den Prozessor im Speicher zu speichern,

wobei die Metadaten, die der ersten Anforderung entsprechen, von Metadaten überschrieben werden, die einer nachfolgenden Anforderung entsprechen,

eine Schreiboperation in demselben Datensatz wie die erste Anforderung durchzuführen.

14. Computerprogrammprodukt nach Anspruch 8, wobei die Prozesse für alle vom Host empfangenen Anforderungen für Schreiboperationen durchgeführt werden.

15. Primäre Datenspeichereinheit, die aufweist: einen Prozessor; und Logik, die in den Prozessor integriert ist, die von dem Prozessor ausführbar ist, oder die in den Prozessor integriert und von dem Prozessor ausführbar ist, wobei die Logik so konfiguriert ist, dass sie: eine erste Anforderung zum Durchführen einer Schreiboperation von einem Host durch den Prozessor empfängt; die Schreiboperation durch den Prozessor durchführt; Metadaten, die der ersten Anforderung entsprechen, durch den Prozessor an die sekundäre Datenspeichereinheit sendet; Metadaten, die einer zweiten Anforderung zum Durchführen der Schreiboperation entsprechen, von der sekundären Datenspeichereinheit durch den Prozessor empfängt, wobei die zweite Anforderung vom Host in der sekundären Datenspeichereinheit empfangen wurde; die Metadaten, die der ersten und zweiten Anforderung entsprechen, durch den Prozessor verwendet, um zu ermitteln, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde; und als Reaktion auf ein Feststellen, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit gespiegelt wurde, eine Antwort an die sekundäre Datenspeichereinheit durch den Prozessor sendet, die anzeigt, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurde.

16. Primäre Datenspeichereinheit nach Anspruch 15, wobei die Logik so konfiguriert ist, dass sie: eine Antwort von der sekundären Datenspeichereinheit durch den Prozessor empfängt, die anzeigt, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurde, als Reaktion auf ein Feststellen, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit gespiegelt wurde, wobei die sekundäre Datenspeichereinheit die Schreiboperation als Reaktion auf Empfangen der zweiten Anforderung zum Durchführen der Schreiboperation durchführt, wobei die sekundäre Datenspeichereinheit die Metadaten verwendet, die der ersten und zweiten Anforderung entsprechen, um zu ermitteln, ob die

Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde.

17. Primäre Datenspeichereinheit nach Anspruch 15, wobei die Logik so konfiguriert ist, dass sie: eine Leseanforderung durch den Prozessor empfängt; durch den Prozessor ermittelt, ob in der Leseanforderung angegebene Daten zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden; als Reaktion auf ein Feststellen, dass die in der Leseanforderung angegebenen Daten nicht zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden, ein Durchführen der Leseanforderung durch den Prozessor verzögert; und als Reaktion auf ein Feststellen, dass die in der Leseanforderung angegebenen Daten zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden, die Leseoperation durch den Prozessor durchführt.

18. Primäre Datenspeichereinheit nach Anspruch 15, wobei die Logik so konfiguriert ist, dass sie: eine Antwort durch den Prozessor an den Host sendet, die anzeigt, dass die Schreiboperation durchgeführt wurde, bevor ermittelt wurde, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurde.

19. Primäre Datenspeichereinheit nach Anspruch 15, wobei die Metadaten, die der ersten Anforderung entsprechen, Informationen umfassen, die aus der Gruppe ausgewählt wurden, die besteht aus: einem Zeitstempel, Hashwerten und einer Speicherortkennung, wobei die Metadaten, die der zweiten Anforderung entsprechen, Informationen umfassen, die aus der Gruppe ausgewählt wurden, die besteht aus: einem Zeitstempel, Hashwerten und einer Speicherortkennung.

20. Primäre Datenspeichereinheit nach Anspruch 15, wobei die Logik so konfiguriert ist, dass sie: die Metadaten, die der ersten Anforderung entsprechen, durch den Prozessor im Speicher speichert, wobei die Metadaten, die der ersten Anforderung entsprechen, von Metadaten überschrieben werden, die einer nachfolgenden Anforderung entsprechen, eine Schreiboperation in demselben Datensatz wie die erste Anforderung durchzuführen.

21. Primäre Datenspeichereinheit nach Anspruch 15, wobei die Prozesse für alle vom Host empfangenen Anforderungen für Schreiboperationen durchgeführt werden.

22. Durch einen Computer implementiertes Verfahren zum Spiegeln von Schreiboperationen zwischen primärer und sekundärer Datenspeichereinheit, das aufweist:

Empfangen einer ersten Anforderung zum Durchführen einer Schreiboperation von einem Host;
Durchführen der Schreiboperation;
Senden von Metadaten, die der ersten Anforderung entsprechen, an die primäre Datenspeichereinheit;
Empfangen von Metadaten, die einer zweiten Anforderung zum Durchführen der Schreiboperation entsprechen, von der primären Datenspeichereinheit, wobei die zweite Anforderung vom Host in der primären Datenspeichereinheit empfangen wurde;
Verwenden der Metadaten, die der ersten und zweiten Anforderung entsprechen, um zu ermitteln, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde; und
als Reaktion auf ein Feststellen, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit gespiegelt wurde, Senden einer Antwort an die primäre Datenspeichereinheit, die anzeigt, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurde.

23. Durch einen Computer implementiertes Verfahren nach Anspruch 22, das aufweist:

Empfangen einer Antwort von der primären Datenspeichereinheit, die anzeigt, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit abgeglichen wurde, als Reaktion auf ein Feststellen, dass die Schreiboperation zwischen der primären und der sekundären Datenspeichereinheit gespiegelt wurde, wobei die primäre Datenspeichereinheit die Schreiboperation als Reaktion auf Empfangen der zweiten Anforderung zum Durchführen der Schreiboperation durchführt, wobei die primäre Datenspeichereinheit die Metadaten verwendet, die der ersten und zweiten Anforderung entsprechen, um zu ermitteln, ob die Schreiboperation zwischen der primären und sekundären Datenspeichereinheit gespiegelt wurde.

24. Durch einen Computer implementiertes Verfahren nach Anspruch 22, das aufweist:

Empfangen einer Leseanforderung;
Ermitteln, ob in der Leseanforderung angegebene Daten zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden;
als Reaktion auf ein Feststellen, dass die in der Leseanforderung angegebenen Daten nicht zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden, Verzögern des Durchführens der Leseanforderung, bis die in der Leseanforderung angegebenen Daten zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden; und

als Reaktion auf ein Feststellen, dass die in der Leseanforderung angegebenen Daten zwischen der primären und sekundären Datenspeichereinheit abgeglichen wurden, Durchführen der Leseoperation.

25. Durch einen Computer implementiertes Verfahren zum Spiegeln von Schreiboperationen zwischen primärer und sekundärer Datenspeichereinheit, das aufweist:

Senden einer ersten Anforderung zum Durchführen einer Schreiboperation an eine primäre Datenspeichereinheit;
Senden einer zweiten Anforderung zum Durchführen der Schreiboperation an eine sekundäre Datenspeichereinheit;
Empfangen einer ersten Antwort von der primären Datenspeichereinheit, die anzeigt, dass die Schreiboperation durchgeführt wurde; und
Empfangen einer ersten Antwort von der primären Datenspeichereinheit, die anzeigt, dass die Schreiboperation durchgeführt wurde.

Es folgen 6 Seiten Zeichnungen

Anhängende Zeichnungen

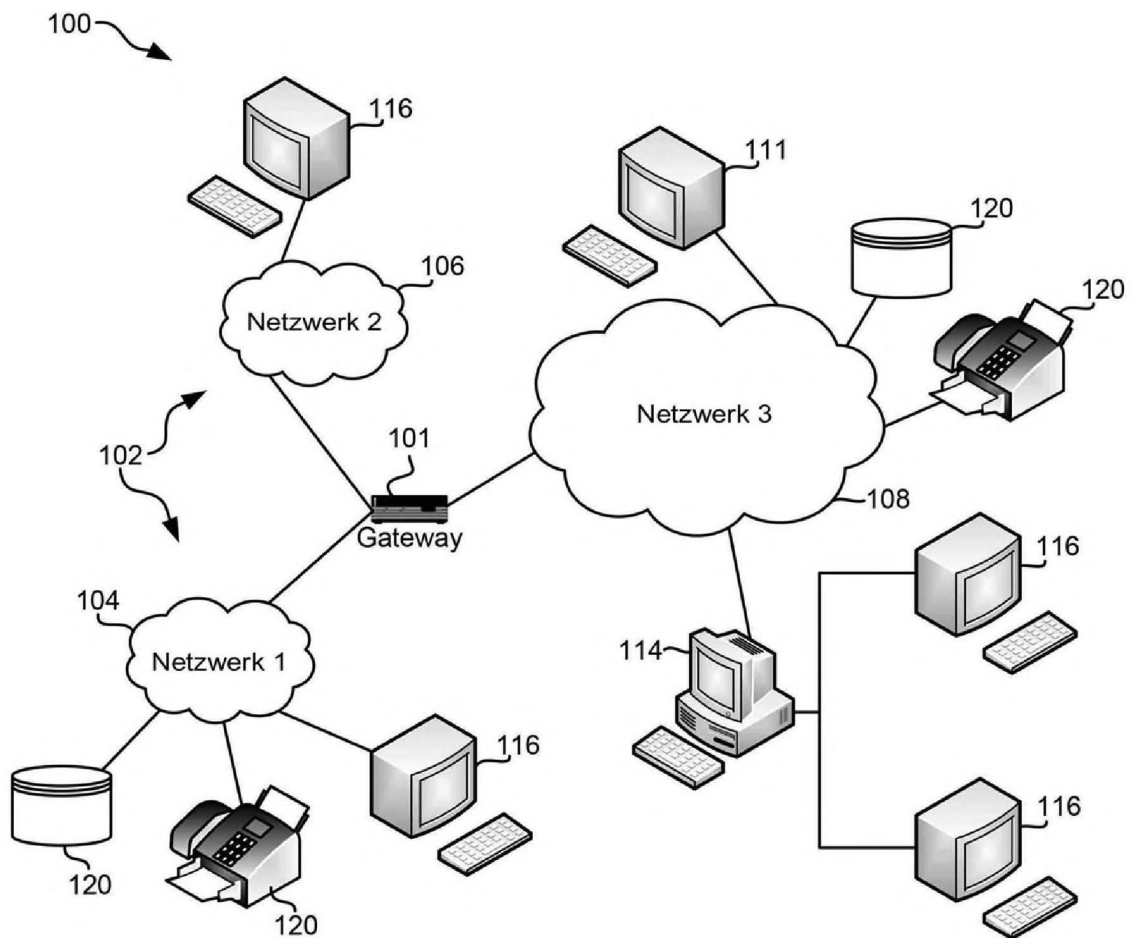


FIG. 1

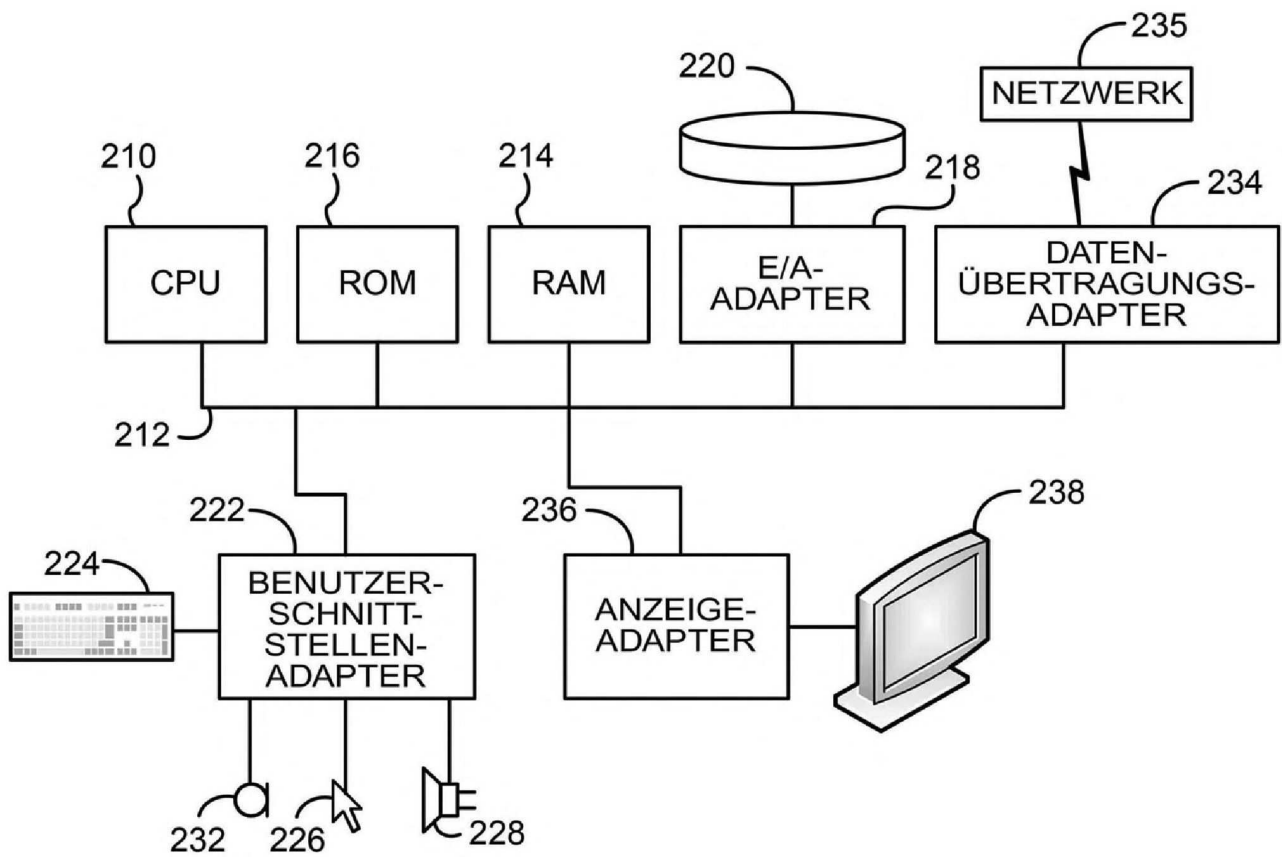


FIG. 2

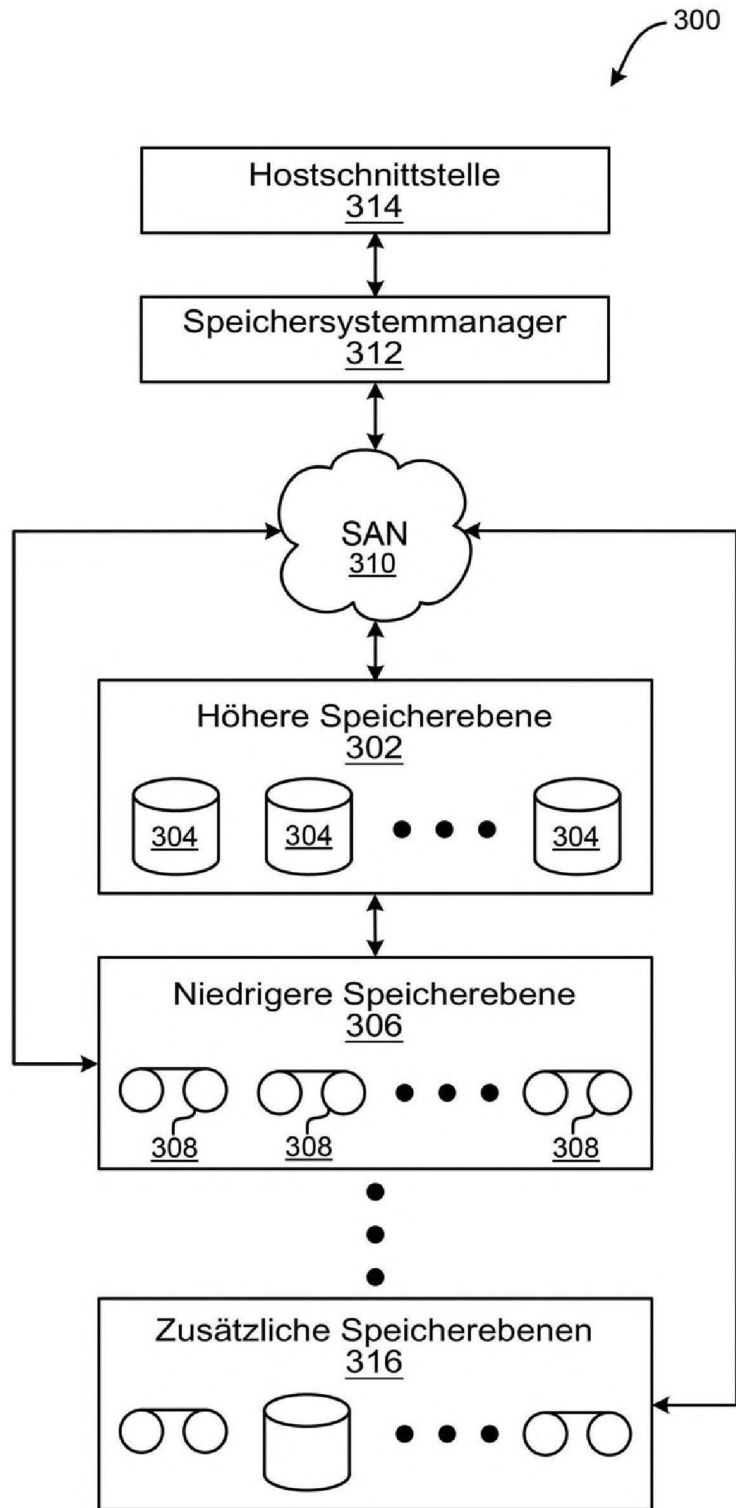


FIG. 3

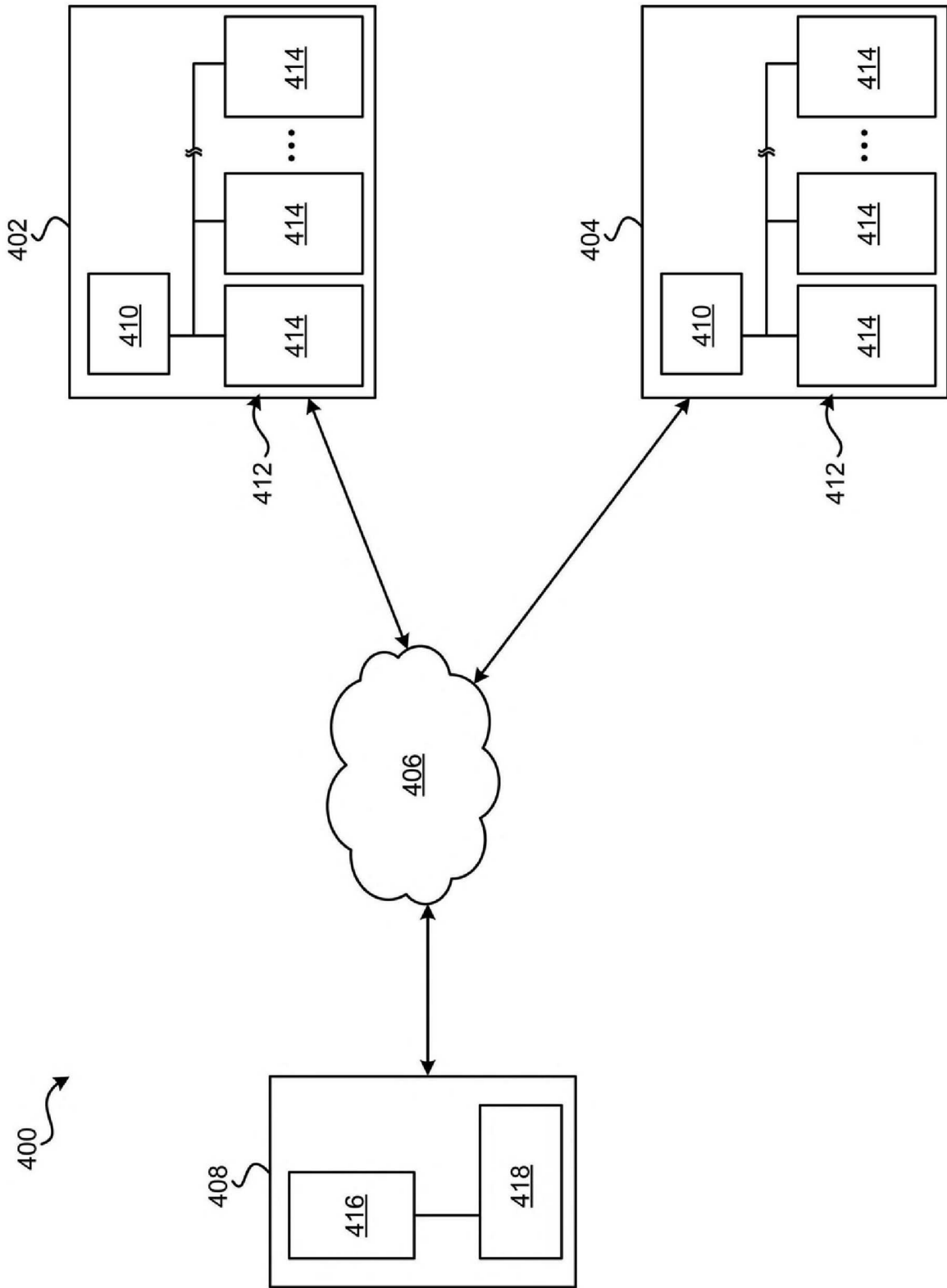


FIG. 4

550 ↘

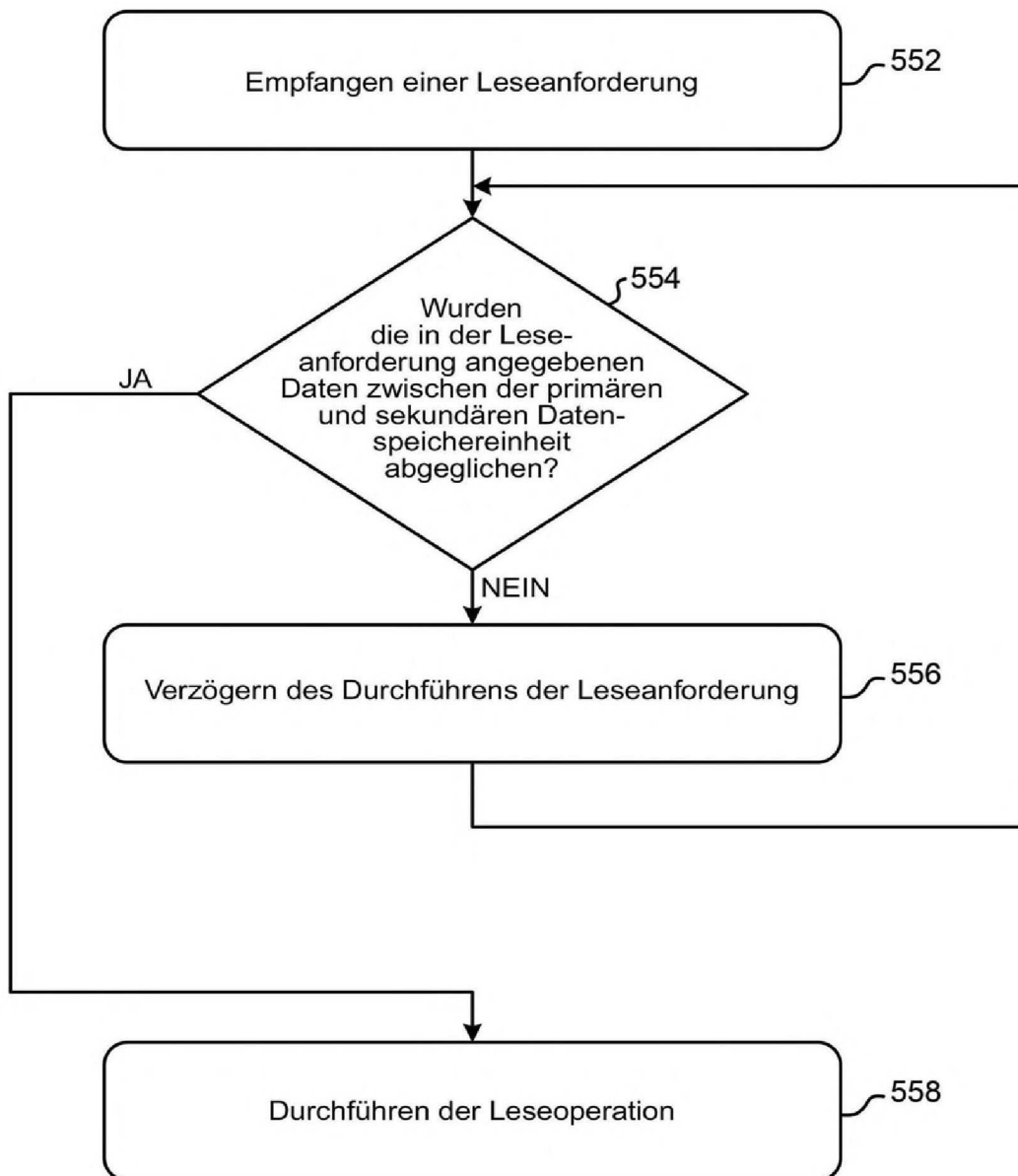


FIG. 5B