



(10) **DE 10 2013 209 528 A1** 2013.12.05

(12) **Offenlegungsschrift**

(21) Aktenzeichen: **10 2013 209 528.7**

(22) Anmeldetag: **23.05.2013**

(43) Offenlegungstag: **05.12.2013**

(51) Int Cl.: **G06F 17/30 (2013.01)**

(30) Unionspriorität:

13/489,383 **05.06.2012** **US**

(71) Anmelder:

**International Business Machines Corp., Armonk,
N.Y., US**

(74) Vertreter:

**RICHARDT PATENTANWÄLTE GbR, 65185,
Wiesbaden, DE**

(72) Erfinder:

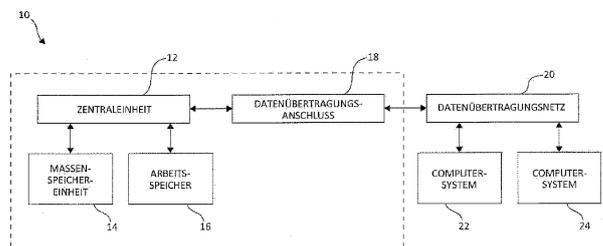
**Akirav, Shay Haim, Tel Aviv, IL; Bachar, Yariv, Tel
Aviv, IL; Levy, Asaf, Tel Aviv, IL; Sonin, Oded, Tel
Aviv, IL; Edelstein, Ron, Tel Aviv, IL**

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Benutzergesteuerte Replikation in einem System für synchronisierte Objektreplicationen**

(57) Zusammenfassung: Eine Momentaufnahme ausgewählter Objekte in einem Quellen-Repository wird als Reaktion auf die benutzergesteuerte Replikation erstellt. Die Momentaufnahme wird zu einem Momentaufnahme-Replikationsjob bestimmt. Der Momentaufnahme-Replikationsjob wird zum Ende einer Replikationswarteschlange hinzugefügt, um die Ausführung für die synchronisierte Objektreplication zu erwarten. Unsynchronisierte Objekte in einem Bestimmungsziel werden durch Vergleichen eines Zustands der ausgewählten Objekte in der Momentaufnahme mit einem aktuellen Zustand des Bestimmungsziels zur Zeit der Ausführung des Momentaufnahme-Replikationsjobs erkannt. Die unsynchronisierten Objekte im Bestimmungsziel werden auf der Grundlage des Vergleichs des Zustands der ausgewählten Objekte in der Momentaufnahme mit dem aktuellen Zustand des Bestimmungsziels zur Zeit der Ausführung des Momentaufnahme-Replikationsjobs synchronisiert.



Beschreibung

HINTERGRUND DER ERFINDUNG

Gebiet der Erfindung

[0001] Die vorliegende Erfindung betrifft allgemein Computer und insbesondere benutzergesteuerte Replikation in einem System für synchronisierte Objektreplication.

Beschreibung des Stands der Technik

[0002] In der heutigen Gesellschaft sind Computersysteme etwas Alltägliches. Computersysteme können am Arbeitsplatz, zuhause oder in der Schule vorhanden sein. Computersysteme können Datenspeichersysteme oder Plattenspeichersysteme enthalten, um Daten zu verarbeiten und zu speichern. Datenreplikation ist ein solcher Prozess. Replizierte Datensysteme können verschiedene logische Datenspeicherentitäten wie Dateien, Datenobjekte, Sicherungs-Abbilder, Daten-Momentaufnahmen oder virtuelle Bandkassetten extern bereitstellen. Häufig ist es erforderlich, dass solche Datenspeicherentitäten von ihrem Ursprungsort an entfernt angeordnete Orte repliziert werden. Replizierte Datenentitäten steigern die Fehlertoleranz-Fähigkeiten und die Verfügbarkeit von Daten. Wenn auch mehrere bestehende Ansätze verwendet werden können, um Datenreplikation durchzuführen, weisen diese Ansätze begleitende Beschränkungen, die sich negativ auf Replikationseffizienz, Systemleistung und Datenkonsistenz auswirken, zusätzlichen Aufwand erfordern, oder irgendeine Kombination der obigen Probleme auf.

KURZBESCHREIBUNG DER ERFINDUNG

[0003] Datenreplikation bedeutet das Sichern von an einem primären Ort gespeicherten Daten durch Speichern des genauen Duplikats der Daten an einem sekundären oder entfernt liegenden Ort. Das Verwenden von Datenreplikationsprozessen bietet verschiedene Vorteile. Datenreplikation trat in den letzten Jahren als ein äußerst wichtiges technologisches Gebiet in Datenverarbeitungs-Speichersystemen in Erscheinung. Probleme mit dem Bereitstellen der Datenreplikations-Funktionalität bestehen in Systemen für synchronisierte Replikation, wo es erforderlich ist, dass strikte Beziehungen zwischen Objekten vorliegen, wie, dass Objekt A mit Zustand X erfordert, dass Objekt B Zustand Y hat. In Systemen für synchronisierte Replikation gibt es beim Erstellen eines neuen Replikationskontexts zwischen der Quelle und dem Ziel oder beim Hinzufügen eines zusätzlichen Satzes von Objekten zum Kontext einen Zustand der Nicht-Synchronisation, da nur neue Operationen an den Objekten an das Ziel gesendet werden. Somit ist es erforderlich, eine Replikation der unsynchronisierten Objekte in ihrem aktuellen Zustand einzuleiten.

[0004] Schlechthin besteht ein Bedürfnis nach einer Lösung, die einen benutzergesteuerten Replikationsprozess in einem System für synchronisierte Objektreplication ermöglicht. Außerdem besteht ein Bedürfnis, die benutzergesteuerte Replikation der unsynchronisierten Objekte durchzuführen, ohne das Erstellen und Bearbeiten der Objekte im Quellen-Repository zu stören, und dabei jede Verletzung der normalen Reihenfolge der Replikation neuer Operationen in der Quelle zu vermeiden. Zusätzlich besteht ein Bedürfnis, einem Benutzer zu gestatten, die Replikation eines Satzes von Objekten ungeachtet des Synchronisationszustands des Satzes von Objekten einzuleiten, um die Objekte, welche eine Replikation erfordern, effizient zu erkennen und die Replikation auf die erkannten, eine Replikation erfordernenden Objekte zu beschränken.

[0005] Demgemäß und in Anbetracht des Vorangehenden werden verschiedene beispielhafte Ausführungsformen von Verfahren, Systemen und Computerprogramm-Produkten für benutzergesteuerte Replikation in einem System für synchronisierte Objektreplication in einer Datenverarbeitungs Umgebung bereitgestellt. In einer Ausführungsform wird, nur als Beispiel, eine Momentaufnahme ausgewählter Objekte in einem Quellen-Repository als Reaktion auf die benutzergesteuerte Replikation erstellt. Die Momentaufnahme wird als ein Momentaufnahme-Replikationsjob ausgewiesen. Der Momentaufnahme-Replikationsjob wird zum Ende einer Replikationswarteschlange hinzugefügt, um die Ausführung für die synchronisierte Objektreplication zu erwarten. Unsynchronisierte Objekte in einem Bestimmungsziel werden durch Vergleichen eines Zustands der ausgewählten Objekte in der Momentaufnahme mit einem aktuellen Zustand des Bestimmungsziels zur Zeit der Ausführung des Momentaufnahme-replikationsjobs erkannt. Die unsynchronisierten Objekte im Bestimmungsziel werden auf der Grundlage des Vergleichs des Zustands der ausgewählten Objekte in der Momentaufnahme mit dem aktuellen Zustand des Bestimmungsziels zur Zeit der Ausführung des Momentaufnahme-replikationsjobs synchronisiert.

[0006] Zusätzlich zur vorangehenden beispielhaften Verfahrens-Ausführungsform werden weitere beispielhafte System- und Computerprodukt-Ausführungsformen bereitgestellt und bieten diese entsprechende Vorteile. Die vorangehende Kurzbeschreibung wurde bereitgestellt, um eine Auswahl von Konzepten in einer vereinfachten Form vorzustellen, welche unten in der ausführlichen Beschreibung weiter beschrieben werden. Diese Kurzbeschreibung soll weder Hauptmerkmale oder wesentliche Merkmale des Anspruchsgegenstands identifizieren noch soll sie als eine Hilfe beim Ermitteln des Umfangs des Anspruchsgegenstands verwendet werden. Der Anspruchsgegenstand ist nicht auf Umsetzungen be-

schränkt, die einigen oder allen im Hintergrund erwähnten Nachteilen abhelfen.

KURZBESCHREIBUNG DER ZEICHNUNGEN

[0007] Damit die Vorteile der Erfindung ohne weiteres verstanden werden, erfolgt eine ausführlichere Beschreibung der oben kurz beschriebenen Erfindung anhand spezieller Ausführungsformen, die in den beigefügten Zeichnungen veranschaulicht werden. Während es sich von selbst versteht, dass diese Zeichnungen Ausführungsformen der Erfindung darstellen und deshalb nicht als ihren Umfang begrenzend zu verstehen sind, wird die Erfindung nun unter Verwendung der folgenden beigefügten Zeichnungen spezieller und ausführlicher beschrieben und erläutert:

[0008] [Fig. 1](#) ist ein Blockschaubild, welches eine Datenverarbeitungssystem-Umgebung mit einer beispielhaften Speichereinheit, worin Aspekte der vorliegenden Erfindung realisiert sein können, veranschaulicht;

[0009] [Fig. 2](#) ist ein Blockschaubild, welches eine Hardwarestruktur eines beispielhaften Datenspeichersystems in einem Computersystem, worin Aspekte der vorliegenden Erfindung realisiert sein können, veranschaulicht;

[0010] [Fig. 3](#) ist ein Ablaufplan, welcher ein beispielhaftes Verfahren zur benutzergesteuerten Replikation in einem System für synchronisierte Objektreplikation veranschaulicht; und

[0011] [Fig. 4](#) ist ein Ablaufplan, welcher ein beispielhaftes Verfahren zur Ausführung eines Momentaufnahmereplikationsjobs veranschaulicht.

AUSFÜHRLICHE BESCHREIBUNG DER ZEICHNUNGEN

[0012] In der folgenden Beschreibung und im Anspruchsgegenstand wird durchweg die folgende, sich auf die veranschaulichten Ausführungsformen beziehende Terminologie verwendet. Ein „Replikationskontext“ soll einen benutzerdefinierten Zusammenhang zwischen einem Satz von Quellenobjekten und einem Ziel bezeichnen. Alle Quellenoperationen an dem Satz von Objekten sind in das Ziel zu replizieren. Ein „Replikationsjob“ soll eine Task, eine Objektoperation von der Quelle (z. B. einem Quellen-Repository) in ein Ziel (z. B. ein Bestimmungsziel) zu replizieren, bezeichnen. Eine „Momentaufnahme“ soll einen interaktiven Zustand eines Objektsatzes zu einem bestimmten Zeitpunkt bezeichnen. Die Momentaufnahme enthält Daten der Objekte und Metadaten zur Zeit der Erstellung der Momentaufnahme. Die Momentaufnahme erhält den Zustand der Objekte aufrecht, selbst wenn sie geändert oder gelöscht

werden. Eine „Replikationswarteschlange“ soll eine Warteschlange bezeichnen, welche die Replikationsjobs enthält, deren Ausführung ansteht. Die Replikationswarteschlange ist eine strikte First-in-first-out-Warteschlange, so dass die Replikationsjobs im Ziel (z. B. im Bestimmungsziel) in genau der gleichen Reihenfolge durchgeführt werden, in welcher sie in der Quelle (z. B. im Quellen-Repository) auftraten. Ein „System“ kann ein Computersystem, einen Computerprozessor, ein primäres Speichersystem, ein entfernt angeordnetes Speichersystem oder jede beliebige sonstige Komponente in einer Computerumgebung, welche zum Ausführen der beanspruchten Erfindung benötigt wird, umfassen.

[0013] Eine „eindeutige Objekt-ID“ soll ein Attribut des Objekts bezeichnen, welches das Objekt innerhalb der Gruppe von Replikationssystemen eindeutig identifiziert. Eine „Objektzustands-ID“ soll ein Attribut des Objektzustands bezeichnen, welches den Zustand des Objekts identifiziert. Die Objektzustands-ID ist für einen bestimmten Zustand eines Objekts eindeutig und ändert sich, wenn die Objektdaten oder Metadaten sich ändern. Eine „Replikations-Engine“ soll das für die physische Replikation von Daten zuständige Modul bezeichnen. Eine „Replikations-Task“ soll eine Anforderung an die Replikations-Engine, die physische Replikation der Objektdaten oder Metadaten durchzuführen, bezeichnen.

[0014] Wie oben erwähnt, bedeutet Datenreplikation das Sichern von an einem primären Ort gespeicherten Daten durch Speichern des genauen Duplikats der Daten an einem sekundären oder entfernt liegenden Ort. Darüber hinaus ist Objektreplikation eher das Verfahren des Replizierens von Objekten auf einer logischen Ebene als das Replizieren auf einer Speicherblockebene. Ähnlich der Replikation eines Speichers auf Blockebene kann Objektreplikation entweder synchronisiert oder unsynchronisiert sein. Im synchronisierten Modus wird jede Operation an den Objekten im Quellen-Repository in das Ziel-Repository gespiegelt. Die Operationen werden im Ziel in der gleichen Reihenfolge durchgeführt, in welcher sie in der Quelle durchgeführt wurden. Im unsynchronisierten Modus besteht keine Notwendigkeit, die Operationen reihenfolgegemäß durchzuführen, sondern lediglich, irgendeinen stabilen Zustand der Objekte im Ziel aufrechtzuerhalten. Ein System für synchronisierte Replikation ist erforderlich, wenn strikte Beziehungen zwischen den Objekten vorliegen, wie dass Objekt A mit Zustand X erfordert, dass Objekt B Zustand Y hat. Zum Beispiel hängen Unterverzeichnisse und/oder Dateien in Systemen für synchronisierte Replikation von übergeordneten Verzeichnissen ab.

[0015] In solchen Systemen gibt es beim Erstellen eines neuen Replikationskontexts zwischen der Quelle und dem Ziel oder beim Hinzufügen eines zu-

sätzlichen Satzes von Objekten zum Kontext einen Zustand der Nicht-Synchronisation, da nur neue Operationen an den Objekten an das Ziel gesendet werden. Somit ist es erforderlich, eine Replikation der unsynchronisierten Objekte in ihrem aktuellen Zustand einzuleiten.

[0016] Schlechthin besteht ein Bedürfnis nach einer Lösung, die einen benutzergesteuerten Replikationsprozess in einem System für synchronisierte Objektreplication ermöglicht. Außerdem besteht ein Bedürfnis, die benutzergesteuerte Replikation der unsynchronisierten Objekte durchzuführen, ohne das Erstellen und Bearbeiten der Objekte im Quellen-Repository zu stören, und dabei jede Verletzung der normalen Reihenfolge der Replikation neuer Operationen in der Quelle zu vermeiden. Zusätzlich besteht ein Bedürfnis, einem Benutzer zu gestatten, die Replikation eines Satzes von Objekten ungeachtet des Synchronisationszustands des Satzes von Objekten einzuleiten, um die Objekte, welche eine Replikation erfordern, effizient zu erkennen und die Replikation auf die erkannten, eine Replikation erfordernden Objekte zu beschränken.

[0017] Um diese Bedürfnisse zu befriedigen, versuchen die Mechanismen der veranschaulichten Ausführungsformen, eine Lösung bereitzustellen, die eine benutzergesteuerte Replikation der unsynchronisierten Objekte gestattet, welche das Erstellen und Bearbeiten der Objekte im Quellen-Repository nicht stört, ohne dabei die normale Reihenfolge der Replikation neuer Operationen in der Quelle zu verletzen. In einer Ausführungsform stellen die Mechanismen, nur als Beispiel, ungeachtet des Synchronisationszustands eines Satzes von Objekten eine benutzergesteuerte Replikation des Satzes von Objekten bereit, erkennen sie effizient die Objekte, welche eine Replikation erfordern, und beschränken sie die Replikation auf lediglich die erkannten, eine Replikation erfordernden Objekte.

[0018] Darüber hinaus sorgen die Mechanismen der vorliegenden Erfindung, nur als Beispiel, für eine skalierbare und effiziente benutzergesteuerte Replikation in einem System für synchronisierte Objektreplication, welches die folgenden wesentlichen Anforderungen erfüllt. 1) Die Mechanismen gestatten dem Benutzer, eine Replikation eines Satzes von Objekten unabhängig von ihrem tatsächlichen Synchronisationszustand durchzuführen. 2) Die Mechanismen erkennen effizient die unsynchronisierten Objekte, welche eine Replikation erfordern. 3) Die Mechanismen stören nicht das lokale Erstellen, Löschen und Bearbeiten von Objekten. 4) Die Mechanismen beeinträchtigen nicht die normale Replikationsreihenfolge der Quellenoperationen. 5) Die Mechanismen gestatten die benutzergesteuerte Replikation eines Satzes von Objekten unabhängig von den eigentlichen Replikationsprozeduren.

[0019] Die Mechanismen der vorliegenden Erfindung erfüllen jeden einzelnen dieser Punkte ungeachtet der Systemgröße und der Anzahl von Objekten. Darüber hinaus ermöglichen die Mechanismen der vorliegenden Erfindung das Einleiten einer Replikation eines Satzes von Objekten ungeachtet ihres Synchronisationszustands, während sie von den Replikationsprozeduren, vom Protokoll und/oder vom Dateisystem-Aufbau, von der Darstellung und der Implementierung unabhängig bleiben.

[0020] In einer Ausführungsform können die Operationen der synchronen Objektreplication der vorliegenden Erfindung, nur als Beispiel, durch das Hinzufügen neuer Quellenoperationen unter dem Replikationskontext als Replikationsjobs zum Ende der Replikationswarteschlange in der Reihenfolge, in welcher sie auftreten, einfach beschrieben werden. Die Mechanismen führen die Jobs aus, indem sie einen Job vom Kopf der Replikationswarteschlange nehmen und die Quellenoperation durchführen, welche der Replikationsjob im Ziel beschreibt. Da die Replikationsjobs in der Reihenfolge, in welcher sie in der Quelle auftreten, zur Replikationswarteschlange hinzugefügt werden und der Reihe nach durchgeführt werden, treten die Replikationsjobs im Ziel in genau der gleichen Reihenfolge auf, in welcher sie in der Quelle auftraten.

[0021] Da der Mechanismus der synchronen Objektreplication nur eine Replikation neuer Operationen an den Objekten durchführt, wenn er den Replikationskontext erstellt oder wenn er diesem zusätzliche Objekte hinzufügt, sind diese Objekte im Ziel nicht vorhanden und werden sie als unsynchronisiert angesehen. Schlechthin sorgen die Mechanismen der vorliegenden Erfindung für eine Lösung, die eine skalierbare und effiziente Einleitung der Replikation eines Objektsatzes innerhalb des grundlegenden Replikationsablaufs ermöglicht, was eine Replikation der Objekte ungeachtet ihres Synchronisationszustands gestattet.

[0022] In einer Ausführungsform erstellen die Mechanismen bei Einleitung der Replikation eines Objektsatzes durch einen Benutzer eine Momentaufnahme des ausgewählten Objektsatzes. Die Mechanismen setzen voraus, dass der zugrundeliegende Aufbau und die zugrundeliegende Darstellung der Objekte einen zügigen Momentaufnahme-Erstellungsprozess unterstützt. Darüber hinaus setzen die Mechanismen voraus, dass der Momentaufnahme-Erstellungsprozess nicht andere lokale Objektoperationen stört. Die Momentaufnahme wird dann als ein neuer Replikationsjob zum Ende der Replikationswarteschlange hinzugefügt. Die Mechanismen der synchronen Replikation gewährleisten die Ausführung der Replikationsjobs in genau der gleichen Reihenfolge, in welcher die Replikationsjobs auftraten, alle Operationen, die vor der Momentaufnahme-

Erstellung auftraten, sind zuverlässig zu replizieren, bevor der Momentaufnahme-Replikationsjob repliziert wird, und alle Operationen, die nach der Momentaufnahme-Replikations-Erstellung auftraten, sind nach der Momentaufnahme-Replikation zu replizieren.

[0023] In **Fig. 1** nun ist eine beispielhafte Architektur **10** einer Datenverarbeitungssystem-Umgebung dargestellt. Das Computersystem **10** enthält eine Zentraleinheit (CPU) **12**, welche mit einem Datenübertragungsanschluss **18** und einer Arbeitsspeichereinheit **16** verbunden ist. Der Datenübertragungsanschluss **18** steht mit einem Datenübertragungsnetz **20** im Datenaustausch. Das Datenübertragungsnetz **20** und ein Speichernetz können so konfiguriert sein, dass sie mit Servern (Hosts) **24** und **22** und Speichersystemen, welche Speichereinheiten **14** enthalten können, im Datenaustausch stehen. Die Speichersysteme können Festplattenlaufwerk-(HDD-(Hard Disk Drive)) Einheiten, Halbleitereinheiten (SSD (Solid-State Devices)) usw. enthalten, welche in einer redundanten Anordnung unabhängiger Festplatten (RAID (Redundant Array of Independent Disks)) konfiguriert sein können. Die Operationen wie unten beschrieben können an Speichereinheiten **14**, welche sich im System **10** oder anderswo befindet bzw. befinden, ausgeführt werden und können mehrere Arbeitsspeichereinheiten **16** aufweisen, die unabhängig und/oder in Verbindung mit weiteren CPU-Einheiten **12** arbeiten. Eine Arbeitsspeichereinheit **16** kann solche Speicher wie elektrisch löschbaren, programmierbaren Nur-Lese-Speicher (EEPROM (Electrically Erasable Programmable Read Only Memory)) oder eine Menge mit entsprechenden Einheiten enthalten. Die Arbeitsspeichereinheit **16** und die Speichereinheiten **14** sind über ein signalführendes Medium mit der CPU **12** verbunden. Zusätzlich ist die CPU **12** über den Datenübertragungsanschluss **18** mit einem Datenübertragungsnetz **20** verbunden, an welches eine Vielzahl von zusätzlichen Computer-Hostsystemen **24** und **22** angeschlossen sind. Zusätzlich können die Arbeitsspeichereinheit **16** und die CPU **12** in jede Komponente des Datenverarbeitungssystems **10** eingebettet und darin enthalten sein. Jedes Speichersystem kann außerdem getrennte und/oder gesonderte Arbeitsspeichereinheiten **16** und eine getrennte und/oder gesonderte CPU **12** enthalten, welche miteinander oder als eine getrennte Arbeitsspeichereinheit **16** und/oder CPU **12** arbeiten.

[0024] **Fig. 2** ist ein beispielhaftes Blockschaubild **200**, welches eine Hardwarestruktur eines Datenspeichersystems in einem Computersystem gemäß der vorliegenden Erfindung zeigt. Host-Computer **210**, **220**, **225** sind gezeigt, die jeweils als eine Zentraleinheit zum Durchführen einer Datenverarbeitung als Teil eines Datenspeichersystems **200** arbeiten. Die Cluster-Hosts/-Knoten (physische oder virtuelle Einheiten) **210**, **220** und **225** können eine oder mehrere neue physische Einheiten oder logische Einheiten

zum Erfüllen der Zwecke der vorliegenden Erfindung im Datenspeichersystem **200** sein. In einer Ausführungsform kann, nur als Beispiel, ein Datenspeichersystem **200** als IBM® System Storage™ DS 8000™ ausgeführt sein. Eine Netzverbindung **260** kann ein Fibre Channel-Fabric, eine Fibre Channel-Punkt-zu-Punkt-Verbindung, ein Fibre Channel-über-Ethernet-Fabric oder eine Fibre Channel-über-Ethernet-Punkt-zu-Punkt-Verbindung, eine FICON- oder ESCON-E/A-Schnittstelle, irgendein anderer E/A-Schnittstellentyp, ein drahtloses Netz, ein Festnetz, ein LAN, ein WAN, heterogen, homogen, öffentlich (d. h. das Internet), privat oder irgendeine Kombination davon sein. Die Hosts **210**, **220** und **225** können lokal oder auf einen oder mehrere Orte verteilt sein und können mit einem beliebigen Fabric-Typ (oder Fabric-Kanal) (in **Fig. 2** nicht gezeigt) oder einem Netzadapter **260** zum Speichercontroller **240** wie einem Fibre Channel-, FICON-, ESCON-, Ethernet-, Lichtwellenleiter-, Mobilnetz- oder Koaxial-Adapter ausgestattet sein. Das Datenspeichersystem **200** ist demgemäß mit einem für den Datenaustausch geeigneten Fabric (in **Fig. 2** nicht gezeigt) oder Netzadapter **260** ausgestattet. Das Datenspeichersystem **200** ist in **Fig. 2** als Speichercontroller **240** und Cluster-Hosts **210**, **220** und **225** aufweisend dargestellt. Die Cluster-Hosts **210**, **220** und **225** können Cluster-Knoten enthalten.

[0025] Um ein deutlicheres Verstehen der hierin beschriebenen Verfahren zu erleichtern, ist der Speichercontroller **240** in **Fig. 2** als eine einzige Verarbeitungseinheit gezeigt, die einen Mikroprozessor **242**, einen Systemarbeitsspeicher **243** und einen nichtflüchtigen Speicher („NVS“ (Nonvolatile Storage)) **216**, welcher unten noch ausführlicher beschrieben wird, aufweist. Besonders zu erwähnen ist, dass der Speichercontroller **240** in einigen Ausführungsformen aus mehreren Verarbeitungseinheiten jeweils mit ihrem eigenen Prozessorkomplex und Systemarbeitsspeicher besteht und über ein dediziertes Netz im Datenspeichersystem **200** angeschlossen ist. Ein Speicher **230** (in **Fig. 3** als **230a**, **230b** und **230n** bezeichnet) kann aus einer oder mehreren Speichereinheiten wie Speicheranordnungen bestehen, welche (über ein Speichernetz) mit dem Speichercontroller **240** verbunden sind, wobei ein oder mehrere Cluster-Hosts **210**, **220** und **225** mit jedem Speichercontroller **240** verbunden sind.

[0026] In einigen Ausführungsformen können die im Speicher **230** enthaltenen Einheiten in einer Ringleitungsarchitektur angeschlossen sein. Der Speichercontroller **240** verwaltet den Speicher **230** und erleichtert die Verarbeitung von für den Speicher **230** bestimmten Schreib- und Leseanforderungen. Der Systemarbeitsspeicher **243** des Speichercontrollers **240** speichert Programmanweisungen und Daten, auf welche der Prozessor **242** zugreifen kann, um Funktionen und Verfahrensschritte der vorliegenden

Erfindung zum Ausführen und Verwalten des Speichers **230**, wie hierin beschrieben, auszuführen. In einer Ausführungsform enthält der Systemarbeitsspeicher **243** die Betriebssoftware **250**, steht er mit ihr in Beziehung oder ist er mit ihr zum Durchführen hierin beschriebener Verfahren und Operationen in Verbindung. Wie in **Fig. 2** gezeigt, kann der Systemarbeitsspeicher **243** auch einen Cache **245** für den Speicher **230**, hierin auch als ein „Cachespeicher“ bezeichnet, zum Puffern von „Schreibdaten“ und „Lesedaten“ enthalten, welche sich auf Schreib- beziehungsweise Leseanforderungen und deren zugehörige Daten beziehen, oder mit diesem in Verbindung sein. In einer Ausführungsform ist der Cachespeicher **245** einer Einheit außerhalb des Systemarbeitsspeichers **243** zugeordnet, bleibt er aber für den Mikroprozessor **242** zugänglich und kann er neben dem Ausführen der Operationen wie hierin beschrieben dazu dienen, zusätzliche Sicherheit gegen Datenverlust bereitzustellen.

[0027] Zur Steigerung der Leistung des Datenspeichersystems **200** ist der Cachespeicher **245** in einigen Ausführungsformen mit einem flüchtigen Speicher und einem nichtflüchtigen Speicher ausgeführt und über einen lokalen Bus (in **Fig. 2** nicht gezeigt) mit dem Mikroprozessor **242** verbunden. Der im Datenspeichercontroller enthaltene NVS **216** ist für den Mikroprozessor **242** zugänglich und dient dazu, zusätzliche Unterstützung für Operationen und Ausführung der vorliegenden Erfindung bereitzustellen, wie in den anderen Figuren beschrieben. Der NVS **216** kann auch als ein „persistenter“ Cache oder „Cachespeicher“ bezeichnet werden und ist mit nichtflüchtigem Speicher ausgeführt, welcher externe Energie nutzen kann, um darin gespeicherte Daten zu bewahren. Der NVS kann zu jedem beliebigen zum Erreichen der Ziele der vorliegenden Erfindung geeigneten Zweck in und mit dem Cachespeicher **245** gespeichert werden. In einigen Ausführungsformen versorgt eine Notstromversorgung (in **Fig. 2** nicht gezeigt) wie eine Batterie den NVS **216** mit genügend Energie, um die darin gespeicherten Daten im Fall eines Stromausfalls im Datenspeichersystem **200** zu bewahren. In bestimmten Ausführungsformen ist die Kapazität des NVS **216** kleiner als die oder gleich der Gesamtkapazität des Cachespeichers **245**.

[0028] Der Speicher **230** kann physisch aus einer oder mehreren Speichereinheiten wie Speicheranordnungen bestehen. Eine Speicheranordnung ist eine logische Gruppierung einzelner Speichereinheiten wie eine Festplatte. In bestimmten Ausführungsformen besteht der Speicher **230** aus einer JBOD-(Just a Bunch of Disks)Anordnung oder einer RAID-(Redundant Array of Independent Disks)Anordnung. Eine Ansammlung physischer Speicheranordnungen kann weiter zusammengefasst werden, um eine Rangfolge zu bilden, was den physischen Speicher von der logischen Konfiguration löst. Der Speicherplatz in

einer Rangfolge kann logischen Datenträgern zugeordnet sein, welche den in einer Schreib-/Leseanforderung angegebenen Speicherort definieren.

[0029] In einer Ausführungsform kann das Speichersystem wie in **Fig. 2** gezeigt, nur als Beispiel, einen logischen Datenträger oder einfach „Datenträger“ enthalten, kann es verschiedene Arten von Zuordnungen aufweisen. Die Speicher **230a**, **230b** und **230n** sind als Rangfolgen im Datenspeichersystem **200** gezeigt und werden hierin als Rang **230a**, **230b** und **230n** bezeichnet. Die Ränge können sich lokal im Datenspeichersystem **200** befinden oder können sich an einem physisch entfernt liegenden Ort befinden. In anderen Worten, ein lokaler Speichercontroller kann mit einem entfernt angeordneten Speichercontroller in Verbindung treten und die Speicherung an dem entfernt liegenden Ort verwalten. Der Rang **230a** ist als mit zwei vollständigen Datenträgern **234** und **236** sowie einem teilweisen Datenträger **232a** konfiguriert gezeigt. Der Rang **230b** ist mit einem weiteren teilweisen Datenträger **232b** gezeigt. Somit ist der Datenträger **232** über die Ränge **230a** und **230b** hinweg zugeordnet. Der Rang **230n** ist als vollständig Datenträger **238** zugeordnet gezeigt – das heißt, der Rang **230n** bezieht sich auf den gesamten physischen Speicher von Datenträger **238**. Aus den obigen Beispielen wird man erkennen, dass ein Rang so konfiguriert sein kann, dass er einen oder mehrere teilweise und/oder vollständige Datenträger enthält. Die Datenträger und Ränge können ferner in sogenannte „Spuren“ unterteilt sein, welche einen festen Speicherblock darstellen. Eine Spur gehört daher zu einem gegebenen Datenträger und kann einen gegebenen Rang erhalten.

[0030] Der Speichercontroller **240** kann ein benutzergesteuertes Replikationsmodul **255**, ein Replikations-Engine-Modul **257** und ein Momentaufnahmemodul **259** enthalten. Das benutzergesteuerte Replikationsmodul **255**, das Replikations-Engine-Modul **257** und das Momentaufnahmemodul **259** können in Verbindung mit jeder einzelnen Komponente des Speichercontrollers **240**, der Hosts **210**, **220**, **225** und der Speichereinheiten **230** arbeiten. Das benutzergesteuerte Replikationsmodul **255**, das Replikations-Engine-Modul **257** und das Momentaufnahmemodul **259** können strukturell ein einziges vollständiges Modul sein oder können zu anderen einzelnen Modulen gehören und/oder darin enthalten sein. Das benutzergesteuerte Replikationsmodul **255**, das Replikations-Engine-Modul **257** und das Momentaufnahmemodul **259** können sich auch im Cachespeicher **245** oder in anderen Komponenten befinden.

[0031] Der Speichercontroller **240** enthält einen Steuerschalter **241** zum Steuern des Fibre Channel-Protokolls für die Host-Computer **210**, **220**, **225**, einen Mikroprozessor **242** zum Steuern des ganzen Speichercontrollers **240**, einen nichtflüchtigen

Steuer-Arbeitsspeicher **243** zum Speichern eines Mikroprogramms (Betriebssoftware) **250** zum Steuern des Betriebs des Speichercontrollers **240**, Daten zur Steuerung und jede später beschriebene Tabelle, den Cachespeicher **245** zum vorübergehenden Speichern (Puffern) von Daten und Puffer **244** zum Unterstützen des Cachespeichers **245**, um Daten zu lesen und zu schreiben, einen Steuerschalter **241** zum Steuern eines Protokolls, um die Datenübertragung zu oder von den Speichereinheiten **230** zu steuern, das benutzergesteuerte Replikationsmodul **255**, das Replikations-Engine-Modul **257** und das Momentaufnahmemodul **259**, in welchem Daten gesetzt werden können. Mehrere Puffer **244** können bei der vorliegenden Erfindung ausgeführt sein, um die Operationen wie hierin beschrieben zu unterstützen. In einer Ausführungsform sind die Cluster-Hosts/-Knoten **210**, **220**, **225** und der Speichercontroller **240** über einen Netzadapter (dies könnte ein Fibre Channel sein) **260** als eine Schnittstelle, d. h. über mindestens einen als „Fabric“ bezeichneten Schalter verbunden.

[0032] In einer Ausführungsform sind die Host-Computer oder eine oder mehrere physische oder virtuelle Einheiten **210**, **220**, **225** und der Speichercontroller **240** über einen Netzadapter (dies könnte ein Fibre Channel sein) **260** als eine Schnittstelle, d. h. über mindestens einen als „Fabric“ bezeichneten Schalter verbunden. In einer Ausführungsform wird die Funktionsweise des in [Fig. 2](#) gezeigten Systems beschrieben. Der Mikroprozessor **242** kann den Arbeitsspeicher **243** so steuern, dass er Befehlsinformationen aus der (physischen oder virtuellen) Host-Einheit **210** und Informationen zum Identifizieren der (physischen oder virtuellen) Host-Einheit **210** speichert. Der Steuerschalter **241**, die Puffer **244**, der Cachespeicher **245**, die Betriebssoftware **250**, der Mikroprozessor **242**, der Arbeitsspeicher **243**, der NVS **216**, das benutzergesteuerte Replikationsmodul **255**, das Replikations-Engine-Modul **257** und das Momentaufnahmemodul **259** stehen miteinander im Datenaustausch und können getrennte Komponenten oder eine einzelne Komponente sein. Außerdem können mehrere, wenn nicht alle Komponenten wie die Betriebssoftware **250** im Arbeitsspeicher **243** enthalten sein. Jede einzelne der Komponenten in den gezeigten Einheiten kann, zu für die vorliegende Erfindung geeigneten Zwecken, mit jeder anderen verbunden sein und im Datenaustausch stehen.

[0033] Wie oben erwähnt, können das benutzergesteuerte Replikationsmodul **255**, das Replikations-Engine-Modul **257** und das Momentaufnahmemodul **259** sich auch im Cachespeicher **245** oder in anderen Komponenten befinden. Besonders erwähnt werden sollte, dass das RAM-Modul **259** mit einem Kurzzeitspeicher und die Festplatte mit dem Langzeitspeicher vergleichbar ist. In anderen Worten, das RAM-Modul **259** ist Arbeitsspeicher. Der RAM (Random Access Memory (Arbeitsspeicher)) ist der Ort

in einem Computer, wo das Betriebssystem, Anwendungsprogramme und Daten, die gerade in Verwendung sind, vorgehalten werden, so dass sie durch den Prozessor **242** des Computers schnell erreicht werden können. Der RAM lässt sich viel schneller auslesen und schreiben als die anderen Arten von Speicher in einem Computer, die Festplatte, die Diskette und die CD-ROM. Schlechthin können ein oder mehrere benutzergesteuerte Replikationsmodule **255**, Replikations-Engine-Module **257** und Momentaufnahmemodule **259** auf der Grundlage von Speicherarchitektur und Benutzervorlieben nach Bedarf verwendet werden.

[0034] [Fig. 3](#) nun ist ein Ablaufplan, welcher ein beispielhaftes Verfahren **300** zur benutzergesteuerten Replikation in einem System für synchronisierte Objektreplication veranschaulicht. Das Verfahren **300** beginnt (Schritt **302**) mit dem Erstellen einer Momentaufnahme ausgewählter Objekte in einem Quellen-Repository als Reaktion auf die benutzergesteuerte Replikation (Schritt **304**). Eine Momentaufnahme wird zu einem Momentaufnahme-Replikationsjob bestimmt (Schritt **306**). Der Momentaufnahme-Replikationsjob wird zum Ende einer Replikationswarteschlange hinzugefügt, um die Ausführung für die synchronisierte Objektreplication zu erwarten (Schritt **308**). Besonders erwähnt werden sollte, dass das Verfahren **300** eine Replikationswarteschlange verwenden kann, indem es den Momentaufnahme-Replikationsjob zur Replikationswarteschlange hinzufügt, um eine Momentaufnahme des vom Benutzer zur Replikation angeforderten Objektsatzes zu replizieren. Der Zustand der ausgewählten Objekte in der Momentaufnahme wird mit einem aktuellen Zustand des Bestimmungsziels zur Zeit der Ausführung des Momentaufnahme-Replikationsjobs verglichen (Schritt **310**). Auf der Grundlage des Vergleichs erkennt das Verfahren **300** unsynchronisierte Objekte in einem Bestimmungsziel (Schritt **312**). Die unsynchronisierten Objekte im Bestimmungsziel werden auf der Grundlage des Vergleichs synchronisiert (Schritt **314**). Das Verfahren **300** endet (Schritt **316**).

[0035] Um die Operationen der vorliegenden Erfindung wie in [Fig. 3](#) beschrieben weiter zu veranschaulichen, betrachten wir die Mechanismen, welche in einem System für synchrone Dateisystem-Replikation verwendet werden. Ein Replikationskontext ist der Zusammenhang zwischen einer Quelle und Zielverzeichnissen. Alle Operationen unter dem Quellenverzeichnis sind im Zielverzeichnis widerzuspiegeln. Ein Objekt in diesem System ist ein Knoten im Baum des Dateisystems, z. B. eine Datei oder ein Verzeichnis. In einer Ausführungsform sind die Objektoperationen in Daten- und Metadatenoperationen aufgeteilt. Die Datenoperationen sind entweder die Schreiboperationen neuer Daten oder die Löschoptionen alter Daten. Die Metadatenoperationen sind entweder

eine Änderung von Metadaten eines Knotens oder die Änderung der Baumstruktur. Solche Operationen werden in der Reihenfolge, in welcher sie auftraten, zur Replikationswarteschlange hinzugefügt und werden im Ziel der Reihe nach ausgeführt. Dadurch gewährleisten die Mechanismen, dass das Ziel immer ein konsistentes Abbild enthält, das einmal in der Quelle vorhanden war.

[0036] Darüber hinaus kann ein Replikationskontext erstellt werden, wenn es bereits vorhandene Objekte in der Quelle gibt. In einem Dateisystem können dies Verzeichnisse mit Dateien sein. Zum Beispiel ist der Replikationskontext die Verbindung zwischen dem Verzeichnis „/a“ im Quellensystem und dem Verzeichnis „/a“ im Zielsystem. Zur Zeit des Erstellens des Replikationskontexts ist das Verzeichnis „/a/b“ in der Quelle bereits einschließlich einer komplexen Unterverzeichnisstruktur darunter vorhanden. Da der Mechanismus der synchronen Replikation nur neue Operationen repliziert, werden der Inhalt und die Struktur von „/a/b“ nicht repliziert, solange sie nicht geändert werden.

[0037] Ein Benutzer fährt dann fort und erstellt das neue Verzeichnis, zum Beispiel „/a/c“, in der Quelle und erstellt eine weitere Unterverzeichnisstruktur darunter. Da diese neuen Operationen unter der Replikationskontext-Überwachung erfolgen, werden sie alle zur Replikationswarteschlange hinzugefügt und werden sie zur Replikation eingeplant. Zu einem bestimmten Zeitpunkt möchte der Benutzer möglicherweise „/a“ vollständig mit dem Ziel synchronisieren. Der Benutzer bewerkstelligt dies durch Einleiten einer Replikation des Verzeichnisses „/a“. Wie in [Fig. 3](#) beschrieben, erstellen die Mechanismen zur Zeit des Einleitens eine Momentaufnahme von „/a“ und fügen sie als einen weiteren Replikationsjob zur Replikationswarteschlange hinzu.

[0038] Zur Zeit des Einleitens ist die Replikationswarteschlange möglicherweise nicht leer, was bedeutet, dass es Quellenoperationen an „/a/c“ gibt, die noch nicht ins Ziel repliziert wurden. In anderen Worten, zur Zeit der Ausführung stimmt der Zielzustand von „/a/c“ nicht mit dem in der Quellen-Momentaufnahme überein. Da die Mechanismen der synchronen Replikation gewährleisten, dass die Replikationsjobs reihenfolgegemäß ausgeführt werden, gewährleisten die Mechanismen, dass bei Ausführung des Momentaufnahmerekopierungsjobs alle früheren Operationen an „/a/c“, die vor seiner Erstellung auftraten, bereits repliziert wurden und dass Operationen an „/a/c“, die nach der Momentaufnahme-Erstellung auftraten, nicht repliziert wurden. In anderen Worten, dies bedeutet, dass das Ziel zur Zeit der Ausführung eine Kopie von „/a/c“ enthält, welche derjenigen in der Momentaufnahme genau gleicht.

[0039] Das Beispielszenario fortführend, arbeiten die Mechanismen durch Vergleichen der Objekte in der Momentaufnahme mit denjenigen im Ziel, was durch Durchlaufen der Momentaufnahme und Vergleichen der Baumstruktur, der Daten und der Metadaten mit dem Ziel geschieht. Darüber hinaus kann der Mechanismus beim Durchführen der Durchlauf- und/oder der Vergleichsoperationen feststellen, dass „/a/c“ vollständig mit der Momentaufnahme übereinstimmt und keine Replikation der Momentaufnahme erforderlich ist. Jedoch fehlen dann alle „/a/b“ im Ziel, so dass die Mechanismen in der Replikations-Engine Replikations-Tasks für die eigentliche Replikation der Daten und Metadaten für alle Knoten in der Momentaufnahme unter „/a/b“ eröffnen können.

[0040] Die Mechanismen warten, bis alle Replikations-Tasks abgeschlossen sind, und dann schließen sie den Momentaufnahme-Replikationsjob ab und gestatten sie, die Quellenoperationen, die danach auftraten, zu replizieren. Dadurch gewährleisten die Mechanismen, dass das Ziel bei Abschluss der Momentaufnahme-Replikation das genaue Abbild von „/a“ zur Zeit der Einleitung einer Replikation durch den Benutzer enthält. Da die Momentaufnahme-Replikation keine Bearbeitungen an „/a/c“ im Ziel vornahm, ist der Zustand von „/a/c“ nicht beeinträchtigt und können die Quellenoperationen in der Warteschlange für „/a/c“ mit dem Replizieren fortfahren.

[0041] In einer Ausführungsform, wie unten in [Fig. 4](#) beschrieben, gestatten die Mechanismen eine benutzergesteuerte Replikation eines Satzes von Objekten durch Erstellen einer Momentaufnahme für den Satz von Objekten, Erstellen eines neuen Replikationsjobs für die Momentaufnahme und Hinzufügen des neuen Replikationsjobs zur Replikationswarteschlange. Die Ausführung des Momentaufnahme-Replikationsjobs erfolgt durch Durchführen von Vergleichen zwischen den Objekten in der Quellen-Momentaufnahme und den Objekten im Ziel und Ausgeben einer Replikation der unsynchronisierten Objekte in die Momentaufnahme, welche eine Replikation erfordern.

[0042] Die Prozedur zur Ausführung des Momentaufnahme-Replikationsjobs wird nun anhand von [Fig. 4](#) beschrieben. [Fig. 4](#) ist ein Ablaufplan, welcher ein beispielhaftes Verfahren **400** zur Ausführung eines Momentaufnahme-Replikationsjobs veranschaulicht. Das Verfahren **400** beginnt (Schritt **402**) mit dem Abrufen des nächsten Objekts aus der Quellen-Momentaufnahme auf eine geordnete Weise (Schritt **404**). Das Verfahren **400** sendet die Objekt-ID und die Zustands-ID des nächsten abgerufenen Objekts an das Bestimmungsziel (Schritt **406**). Das Verfahren **400** ermittelt, ob die Objekt-ID im Bestimmungsziel vorhanden ist (Schritt **408**). Falls nein, kann das Verfahren **400** Replikations-Tasks eröffnen, um die Objektdaten und Metadaten zu replizieren (Schritt **412**). Falls ja, ermittelt das Verfahren **400**, ob die empfan-

gene Zustands-ID (z. B. vom nächsten ausgewählten Objekt empfangen) von der aktuellen Zustands-ID im Bestimmungsziel verschieden ist (Schritt 410). Falls ja, eröffnet das Verfahren 400 Replikations-Tasks, um die Objektdaten und Metadaten zu replizieren (Schritt 412). Falls nein, ermittelt das Verfahren 400, ob das Ende der Momentaufnahme erreicht wurde (Schritt 414). Falls ja, wartet das Verfahren 400, bis alle Replikations-Tasks abgeschlossen sind (Schritt 416). Falls nein, kehrt das Verfahren 400 zu Schritt 404 zurück und ruft es das nächste Objekt aus der Quellen-Momentaufnahme ab (Schritt 404). Wieder in Schritt 412, fährt das Verfahren 400 nach Abschluss des Eröffnens von Replikations-Tasks zum Replizieren der Objektdaten und der Metadaten mit Schritt 414 fort. Wiederum ermittelt das Verfahren 400, ob das Ende der Momentaufnahme erreicht wurde (Schritt 414). Falls ja, wartet das Verfahren 400, bis alle Replikations-Tasks abgeschlossen sind (Schritt 416). Falls nein, kehrt das Verfahren 400 zu Schritt 404 zurück und ruft es das nächste Objekt aus der Quellen-Momentaufnahme ab (Schritt 404). Das Verfahren 400 endet (Schritt 418).

[0043] Bei Ausführung des Momentaufnahme-Replikationsjobs wird, wie oben in Fig. 4 beschrieben, der Zustand der in der Momentaufnahme enthaltenen Objekte mit ihrem aktuellen Zustand im Ziel verglichen. Die Mechanismen führen den Vergleich an einer schlanken Darstellung der Objekte (z. B. Version und/oder Hashwert) und nicht an den vollständigen Objektdaten durch, wodurch sie eine skalierbare und effiziente Operation ermöglichen. Die Mechanismen entdecken die Zielobjekte, die nicht synchronisiert sind, und aktualisieren sie gemäß dem Zustand in der Quellen-Momentaufnahme.

[0044] In einer Ausführungsform haben die Mechanismen zur Momentaufnahme-Replikation besondere Eigenschaften, die eine skalierbare und effiziente benutzergesteuerte Replikation in einem System für synchronisierte Replikation ermöglichen, welche der umfassenden, wie folgt zusammengesetzten Liste von Anforderungen genügt. 1) Die Mechanismen gestatten dem Benutzer, eine Replikation eines Satzes von Objekten unabhängig von ihrem tatsächlichen Synchronisationszustand durchzuführen. 2) Die Mechanismen erkennen effizient unsynchronisierte Objekte, indem sie ihren Zustand im Ziel mit ihrem Zustand in der Quellen-Momentaufnahme vergleichen. 3) Die Mechanismen der vorliegenden Erfindung stören nicht das lokale Erstellen und Bearbeiten von Objekten, da die Mechanismen nur eine Momentaufnahme des Zustands der Objekte erstellen, wobei sie den vom logischen Speichersystem bereitgestellten effizienten Momentaufnahme-Mechanismus verwenden. 4) Die Mechanismen beeinträchtigen nicht die normale Replikationsreihenfolge der Quellenoperationen, da eine Momentaufnahme als ein weiterer Replikationsjob zur Replikation hinzugefügt wird. 5)

Die Mechanismen gestatten für die benutzergesteuerte Replikation eines Satzes von Objekten unabhängig von den eigentlichen Replikationsprozeduren zu sein.

[0045] Wie der Fachmann erkennen wird, können Aspekte der vorliegenden Erfindung als ein System, ein Verfahren oder ein Computerprogramm-Produkt verkörpert sein. Entsprechend können Aspekte der vorliegenden Erfindung die Form einer vollständig auf Hardware beruhenden Ausführungsform, einer vollständig auf Software beruhenden Ausführungsform (darunter Firmware, residente Software, Mikrocode usw.) oder einer Software- und Hardware-Aspekte, die hierin alle allgemein als „Schaltung“, „Modul“ oder „System“ bezeichnet werden können, kombinierenden Ausführungsform annehmen. Überdies können Aspekte der vorliegenden Erfindung die Form eines Computerprogramm-Produkts annehmen, das in einem oder mehreren computerlesbaren Medien, auf denen computerlesbarer Programmcode enthalten ist, verkörpert ist.

[0046] Jede beliebige Kombination von einem oder mehreren computerlesbaren Medien kann verwendet werden. Bei dem computerlesbaren Medium kann es sich um ein computerlesbares Signalmedium oder ein computerlesbares Speichermedium handeln. Zu einem computerlesbaren Speichermedium können zum Beispiel, ohne darauf beschränkt zu sein, ein elektronisches, magnetisches, optisches, elektromagnetisches, Infrarot- oder Halbleiter-System, eine elektronische, magnetische, optische, elektromagnetische, Infrarot- oder Halbleiter-Vorrichtung oder -Einheit oder irgendeine geeignete Kombination der Vorerwähnten gehören. Zu spezielleren Beispielen (eine nichterschöpfende Liste) des computerlesbaren Speichermediums würde Folgendes zählen: eine elektrische Verbindung mit einer oder mehreren Leitungen, eine tragbare Computerdiskette, eine Festplatte, ein Direktzugriffsspeicher (RAM, Random Access Memory), ein Festwertspeicher (ROM, Read-Only Memory), ein löschbarer programmierbarer Festwertspeicher (EPROM, Erasable Programmable Read-Only Memory, oder Flash-Speicher), ein Lichtwellenleiter, ein tragbarer Compact-Disc-Festwertspeicher (CD-ROM), eine Einheit zur optischen Speicherung, eine Einheit zur magnetischen Speicherung oder irgendeine geeignete Kombination der Vorerwähnten. Im Kontext dieses Dokuments kann ein computerlesbares Speichermedium irgendein materielles Medium sein, das ein Programm zur Verwendung durch oder in Verbindung mit ein bzw. einem Befehlsausführungssystem, eine bzw. einer Befehlsausführungsvorrichtung oder eine bzw. einer Befehlsausführungseinheit enthalten oder speichern kann.

[0047] Auf einem computerlesbaren Medium enthaltener Programmcode kann mittels irgendeines

geeigneten Mediums einschließlich, aber nicht beschränkt auf drahtlos, drahtgebunden, Lichtwellenleiter, HF usw. oder irgendeiner geeigneten Kombination der Vorerwähnten übertragen werden. Computerprogrammcode zum Ausführen von Operationen für Aspekte der vorliegenden Erfindung kann in irgendeiner Kombination einer oder mehrerer Programmiersprachen, darunter objektorientierte Programmiersprachen wie Java, Smalltalk, C++ oder dergleichen und herkömmliche verfahrensorientierte Programmiersprachen wie die Programmiersprache „C“ oder ähnliche Programmiersprachen, geschrieben sein. Der Programmcode kann ganz 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 ganz auf dem entfernt angeordneten Computer oder Server ausgeführt werden. Im letzteren Szenario kann der entfernt angeordnete Computer über jede Art von Netzwerk, darunter ein lokales Netz (LAN, Local Area Network) oder ein weiträumiges Netz (WAN, Wide Area Network), mit dem Computer des Benutzers verbunden sein oder kann die Verbindung zu einem externen Computer hergestellt sein (zum Beispiel mittels eines Internet-Diensteanbieters über das Internet).

[0048] Aspekte der vorliegenden Erfindung wurden oben unter Bezugnahme auf Ablaufpläne und/oder Blockschaubilder von Verfahren, Vorrichtungen (Systemen) und Computerprogramm-Produkten gemäß Ausführungsformen der Erfindung beschrieben. Es versteht sich von selbst, dass jeder Block der Ablaufpläne und/oder Blockschaubilder und Kombinationen von Blöcken in den Ablaufplänen und/oder Blockschaubildern durch Computerprogrammbeefehle implementiert werden kann. Diese Computerprogrammbeefehle können an einen Prozessor eines Universalcomputers, eines Spezialcomputers oder einer anderen programmierbaren Datenverarbeitungsvorrichtung gegeben werden, um eine Maschine zu produzieren, so dass die Befehle, welche über den Prozessor des Computers oder einer anderen programmierbaren Datenverarbeitungsvorrichtung ausgeführt werden, ein Mittel zum Implementieren der im Block oder in den Blöcken des Ablaufplans und/oder des Blockschaubilds angegebenen Funktionen/Vorgänge schaffen.

[0049] Diese Computerprogrammbeefehle können auch in einem computerlesbaren Medium gespeichert sein, das einen Computer, eine andere programmierbare Datenverarbeitungsvorrichtung oder andere Einheiten anweisen kann, auf eine bestimmte Weise zu funktionieren, so dass die im computerlesbaren Medium gespeicherten Befehle ein Erzeugnis produzieren, das Befehle enthält, welche die bzw. den im Block oder in den Blöcken von Ablaufplan und/oder Blockschaubildern angegebene

(n) Funktion/Vorgang implementieren. Die Computerprogrammbeefehle können auch auf einen Computer, eine andere programmierbare Datenverarbeitungsvorrichtung oder andere Einheiten geladen werden, um zu veranlassen, dass eine Reihe von Arbeitsschritten auf dem Computer, einer anderen programmierbaren Vorrichtung oder den anderen Einheiten ausgeführt werden, um einen computerausgeführten Prozess zu produzieren, so dass die Befehle, welche auf dem Computer oder einer anderen programmierbaren Vorrichtung ausgeführt werden, Prozesse zum Umsetzen der im Block oder in den Blöcken von Ablaufplan und/oder Blockschaubildern angegebenen Funktionen/Vorgänge schaffen.

[0050] Die Ablaufdiagramme und die Blockschaubilder in den obigen Figuren veranschaulichen Architektur, Funktionalität und Betrieb möglicher Umsetzungen von Systemen, Verfahren und Computerprogramm-Produkten gemäß verschiedenen Ausführungsformen der vorliegenden Erfindung. In dieser Hinsicht kann jeder Block im Ablaufplan oder in den Blockschaubildern ein Codemodul, einen Codeabschnitt oder einen Codeteil darstellen, welches bzw. welcher einen oder mehrere lauffähige Befehle zum Umsetzen der angegebenen logischen Verknüpfungen) enthält. Es sollte auch bemerkt werden, dass bei manchen alternativen Umsetzungen die im Block angegebenen Funktionen außerhalb der in den Figuren angegebenen Reihenfolge auftreten können. Zum Beispiel können zwei aufeinanderfolgend gezeigte Blöcke tatsächlich im Wesentlichen gleichzeitig ausgeführt werden oder können die Blöcke manchmal in umgekehrter Reihenfolge ausgeführt werden, je nach erforderlicher Funktionalität. Es wird auch bemerkt werden, dass jeder Block der Blockschaubilder und/oder des Ablaufplans und Kombinationen von Blöcken in den Blockschaubildern und/oder im Ablaufplan durch spezielle, auf Hardware beruhende Systeme, welche die angegebenen Funktionen oder Vorgänge ausführen, oder Kombinationen von Spezialhardware und Computerbefehlen ausgeführt werden können.

[0051] Wenn auch eine oder mehrere Ausführungsformen der vorliegenden Erfindung eingehend dargestellt wurden, wird der Fachmann erkennen, dass Veränderungen und Anpassungen an diesen Ausführungsformen vorgenommen werden können, ohne vom Umfang der vorliegenden Erfindung, wie in den folgenden Ansprüchen bekanntgemacht, abzuweichen.

Patentansprüche

1. Verfahren zur benutzergesteuerten Replikation in einem System für synchronisierte Objektreplikation in einer eine Prozessoreinheit verwendenden Datenverarbeitungsumgebung, welches aufweist:

das Erstellen einer Momentaufnahme ausgewählter Objekte in einem Quellen-Repository als Reaktion auf die benutzergesteuerte Replikation, wobei die Momentaufnahme zu einem Momentaufnahme-Replikationsjob bestimmt wird;
 das Hinzufügen des Momentaufnahme-Replikationsjobs zu einem Ende einer Replikationswarteschlange, um die Ausführung für die synchronisierte Objektreplication zu erwarten; und
 das Erkennen unsynchronisierter Objekte in einem Bestimmungsziel durch Vergleichen eines Zustands der ausgewählten Objekte in der Momentaufnahme mit einem aktuellen Zustand des Bestimmungsziels zur Zeit der Ausführung des Momentaufnahme-Replikationsjobs, wobei die unsynchronisierten Objekte im Bestimmungsziel auf der Grundlage des Vergleichs des Zustands der ausgewählten Objekte in der Momentaufnahme mit dem aktuellen Zustand des Bestimmungsziels zur Zeit der Ausführung des Momentaufnahme-Replikationsjobs synchronisiert werden.

2. Verfahren nach Anspruch 1, außerdem aufweisend das Ausführen entweder von:

dem Replizieren der Operation vor dem Replizieren des Momentaufnahme-Replikationsjobs, wenn eine Operation vor dem Erstellen auftritt, oder,
 dem Replizieren der Operation nach dem Replizieren des Momentaufnahme-Replikationsjobs, wenn eine Operation nach dem Erstellen auftritt,.

3. Verfahren nach Anspruch 2, außerdem aufweisend das Aufteilen der Operation in eine Datenoperation und eine Metadatenoperation, wobei die Datenoperation entweder aus dem Schreiben neuer Daten oder aus dem Löschen alter Daten besteht und die Metadatenoperation entweder aus dem Ändern von Metadaten eines Knotens oder dem Ändern einer Baumstruktur besteht.

4. Verfahren nach Anspruch 1, wobei das Vergleichen eher das Vergleichen einer schlanken Darstellung der ausgewählten Objekte als aller Daten in einem Satz der ausgewählten Objekte aufweist.

5. Verfahren nach Anspruch 1, außerdem aufweisend, in Verbindung mit dem Vergleich, das Durchlaufen der Momentaufnahme durch Vergleichen mindestens eines von einer Baumstruktur und/oder von Daten und/oder von Metadaten der Momentaufnahme mit Objekten im Bestimmungsziel.

6. Verfahren nach Anspruch 1, außerdem aufweisend, in Verbindung mit der Ausführung des Momentaufnahme-Replikationsjobs, das Ausführen entweder von:
 dem Abrufen mindestens eines Objekts der ausgewählten Objekte aus der Momentaufnahme auf eine geordnete Weise,

dem Senden einer Objektidentifikation (ID) und einer Zustands-ID des mindestens einen Objekts an das Bestimmungsziel,
 dem Suchen nach der Objekt-ID im Bestimmungsziel, wobei, wenn die Objekt-ID ausfindig gemacht ist, die Zustands-ID des mindestens einen Objekts mit einer aktuellen Zustands-ID des Bestimmungsziels verglichen wird, oder,
 wenn weder die Objekt-ID im Bestimmungsziel vorhanden ist noch die Zustands-ID des mindestens einen Objekts und die aktuelle Zustands-ID des Bestimmungsziels übereinstimmen, werden Replikations-Tasks zum Replizieren von Daten und Metadaten des mindestens einen Objekts in das Bestimmungsziel geöffnet.

7. Verfahren nach Anspruch 1 außerdem aufweisend, in Verbindung mit der benutzergesteuerten Replikation der ausgewählten Objekte, das Replizieren der ausgewählten Objekte unabhängig von einem tatsächlichen synchronisierten Zustand der ausgewählten Objekte.

8. Verfahren nach Anspruch 1, außerdem aufweisend, in Verbindung mit der benutzergesteuerten Replikation der ausgewählten Objekte, das Replizieren der ausgewählten Objekte unabhängig von mindestens einem von einer Replikationsprozedur und/oder einem Protokoll und/oder einem Dateisystem-Aufbau und/oder einer Implementierung.

9. System zur benutzergesteuerten Replikation in einem System für synchronisierte Objektreplication in einer Datenverarbeitungsumgebung, welches aufweist:

eine in der Datenverarbeitungsumgebung betreibbare Prozessoreinheit, wobei die Prozessoreinheit geeignet ist für:

das Erstellen einer Momentaufnahme ausgewählter Objekte in einem Quellen-Repository als Reaktion auf die benutzergesteuerte Replikation, wobei die Momentaufnahme zu einem Momentaufnahme-Replikationsjob bestimmt wird,
 das Hinzufügen des Momentaufnahme-Replikationsjobs zu einem Ende einer Replikationswarteschlange, um die Ausführung für die synchronisierte Objektreplication zu erwarten, und
 das Erkennen unsynchronisierter Objekte in einem Bestimmungsziel durch Vergleichen eines Zustands der ausgewählten Objekte in der Momentaufnahme mit einem aktuellen Zustand des Bestimmungsziels zur Zeit der Ausführung des Momentaufnahme-Replikationsjobs, wobei
 die unsynchronisierten Objekte im Bestimmungsziel auf der Grundlage des Vergleichs des Zustands der ausgewählten Objekte in der Momentaufnahme mit dem aktuellen Zustand des Bestimmungsziels zur Zeit der Ausführung des Momentaufnahme-Replikationsjobs synchronisiert werden.

10. Computerprogramm-Produkt für benutzergesteuerte Replikation in einem System für synchronisierte Objektreplication in einer eine Prozessoreinheit verwendenden Datenverarbeitungsumgebung, wobei das Computerprogramm-Produkt ein computerlesbares Speichermedium mit darin gespeicherten computerlesbaren Programmcodeabschnitten aufweist, wobei die computerlesbaren Programmcodeabschnitte aufweisen:

einen ersten ausführbaren Abschnitt zum Erstellen einer Momentaufnahme ausgewählter Objekte in einem Quellen-Repository als Reaktion auf die benutzergesteuerte Replikation, wobei die Momentaufnahme zu einem Momentaufnahme-Replikationsjob bestimmt wird;

einen zweiten ausführbaren Abschnitt zum Hinzufügen des Momentaufnahme-Replikationsjobs zu einem Ende einer Replikationswarteschlange, um die Ausführung für die synchronisierte Objektreplication zu erwarten; und

einen dritten ausführbaren Abschnitt zum Erkennen unsynchronisierter Objekte in einem Bestimmungsziel durch Vergleichen eines Zustands der ausgewählten Objekte in der Momentaufnahme mit einem aktuellen Zustand des Bestimmungsziels zur Zeit der Ausführung des Momentaufnahme-Replikationsjobs, wobei die unsynchronisierten Objekte im Bestimmungsziel auf der Grundlage des Vergleichs des Zustands der ausgewählten Objekte in der Momentaufnahme mit dem aktuellen Zustand des Bestimmungsziels zur Zeit der Ausführung des Momentaufnahme-Replikationsjobs synchronisiert werden.

Es folgen 4 Blatt Zeichnungen

Anhängende Zeichnungen

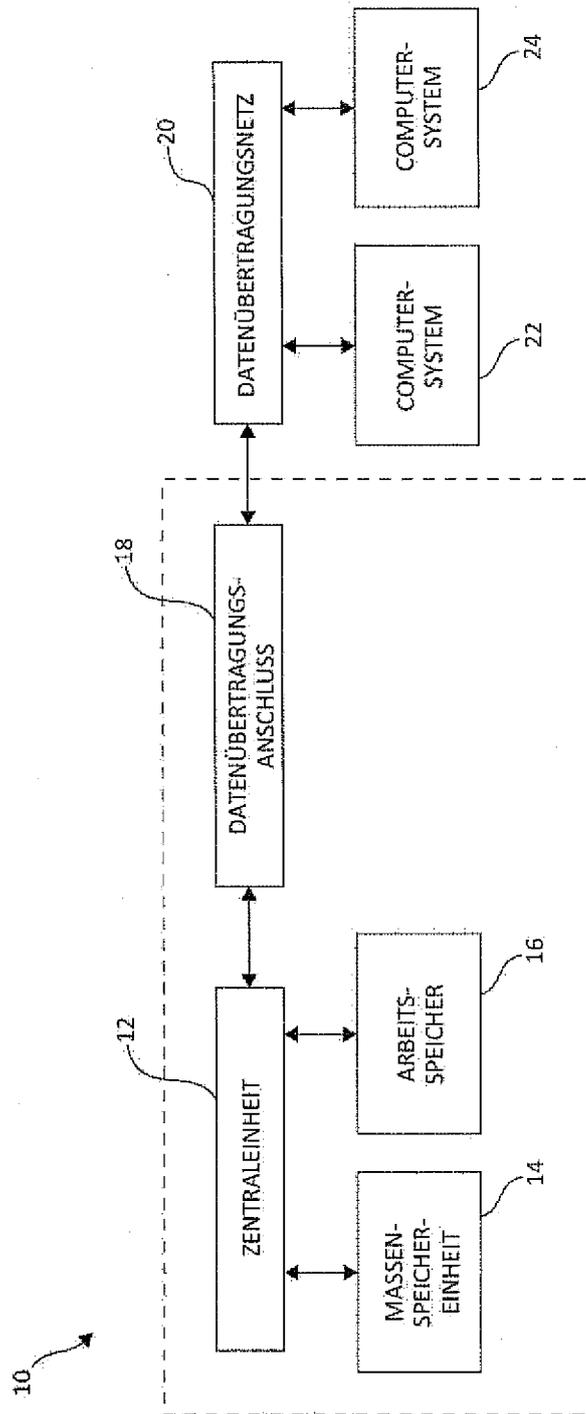


FIG. 1

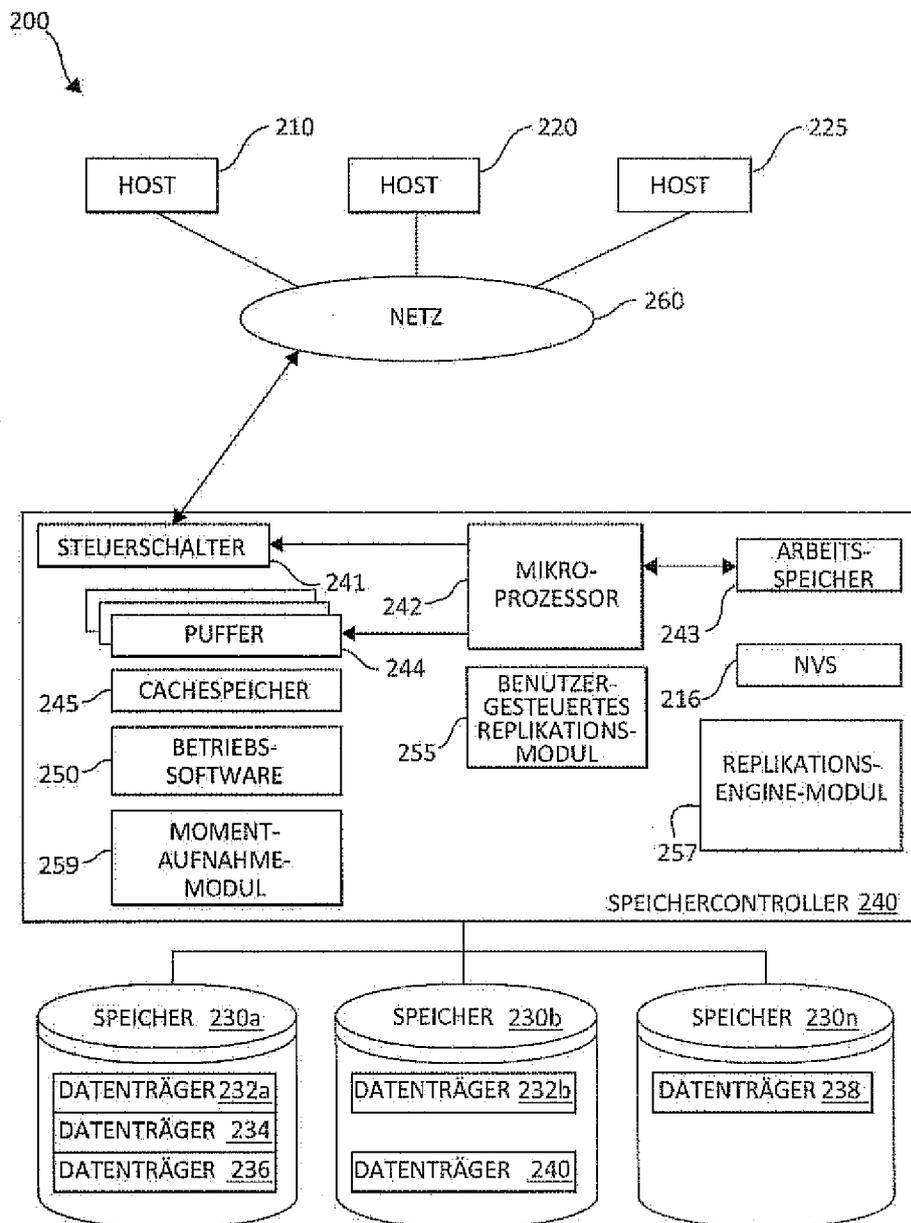


FIG. 2

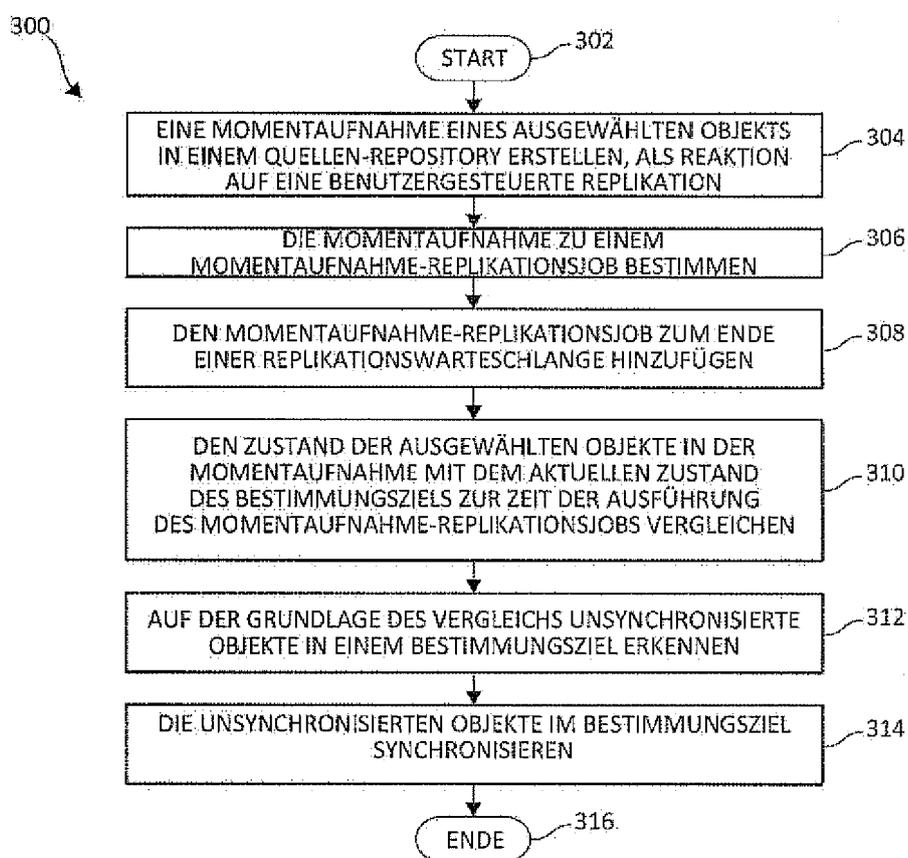


FIG. 3

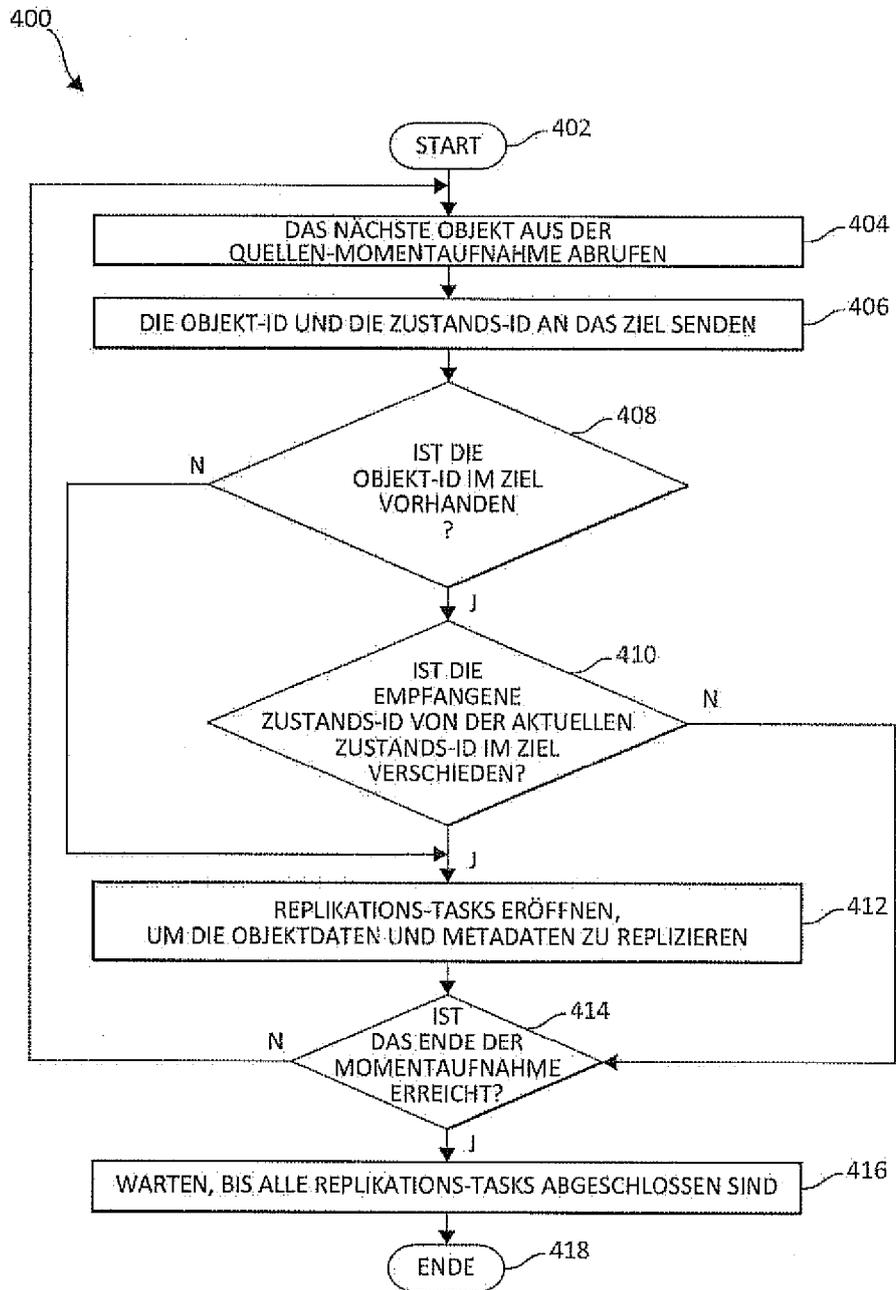


FIG. 4