(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 697 22 962 T2** 2004.05.19

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 978 069 B1**
(21) Deutsches Aktenzeichen: **697 22 962.9**
(86) PCT-Aktenzeichen: **PCT/US97/21466**
(96) Europäisches Aktenzeichen: **97 949 572.8**
(87) PCT-Veröffentlichungs-Nr.: **WO 98/022892**
(86) PCT-Anmeldetag: **21.11.1997**
(87) Veröffentlichungstag der PCT-Anmeldung: **28.05.1998**
(97) Erstveröffentlichung durch das EPA: **09.02.2000**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **18.06.2003**
(47) Veröffentlichungstag im Patentblatt: **19.05.2004**

(51) Int Cl.[7]: **G06F 17/30**
G06F 12/10

(30) Unionspriorität:

754481	**22.11.1996**	**US**
827534	**28.03.1997**	**US**

(84) Benannte Vertragsstaaten:
AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LI, NL, PT, SE

(73) Patentinhaber:
MangoSoft Corp., Westborough, Mass., US

(74) Vertreter:
Hössle Kudlek & Partner, Patentanwälte, 70184 Stuttgart

(72) Erfinder:
DAVIS, H., Scott, Groton, US; CARTER, B., John, Salt Lake City, US; FRANK, J., Steven, Hopkinton, US; DIETTERICH, J., Daniel, Acton, US; LEE, H., Hsin, Acton, US

(54) Bezeichnung: **STRUKTURIERTES DATENSPEICHERSYSTEM MIT GLOBAL ADRESSIERBAREM SPEICHER**

Beschreibung

[0001] Die vorliegende Erfindung betrifft allgemein strukturierte Speichersysteme (beispielsweise Dateisysteme, Datenbanksysteme und Systeme zum Speichern, Gemeinsamverwenden und Ausliefern von Datenobjekten, JAVA-Applets und Web-Seiten). Die Erfindung betrifft insbesondere Systeme und Verfahren, die einen strukturierten Datenspeicher, vorzugsweise innerhalb eines verteilten, adressierbaren, gemeinsam verwendeten Speicherplatzes, erhalten.

[0002] Computerbasierte strukturierte Speichersysteme, wie Computerdateisysteme und Datenbanksysteme, waren bemerkenswert erfolgreich, Benutzern einen schnellen und einfachen Zugriff auf enorme Datenmengen zu geben. Die Wichtigkeit dieser strukturierten Speichersysteme im heutigen Handel kann kaum übertrieben werden. Beispielsweise haben es strukturierte Speichersysteme Gesellschaften ermöglicht, enorme Speicher permanenter Daten, die die Firma modifizieren und aktualisieren kann, im Laufe der Jahre zu erzeugen und zu unterhalten. Für viele Firmen sind diese permanenten Daten ein wertvolles Kapital, das täglich verwendet wird, um die Kernoperationen der Firma auszuführen. Die Daten können beispielsweise Computerdateien (beispielsweise Quellcode, Textverarbeitungsdokumente usw.), Datenbank-Datensätze und Informationen (beispielsweise Informationen zu Angestellten, Kunden und/oder Produkten) und/oder Web-Seiten sein.

[0003] Ein typisches computerbasiertes strukturiertes Speichersystem weist einen zentralen Server, wie bspw. ein Dateisystem-Server oder ein Datenbanksystem-Server, auf, der eine zentralisierte Steuerung des strukturierten Datenspeichers bereitstellt. Der strukturierte Datenspeicher besteht aus den Informationen, die von dem System unterhalten werden, wie den Informationen in den Dateien und den Verzeichnissen des Dateisystems oder den Informationen innerhalb der Zeilen und Spalten der Tabellen des Datenbanksystems. Der zentrale Server stellt einer Mehrzahl miteinander verbundener Netzwerk-Clientknoten Systemdienste bereit, und jeder der Clientknoten verwendet den zentralen Server zum Zugreifen auf den strukturierten Datenspeicher und zum Manipulieren von diesem. Dementsprechend stellt der zentrale Server einen Kern für das strukturierte Speichersystem bereit und erhält eine zentrale Steuerung des Systems und der darin gespeicherten Daten aufrecht.

[0004] Wenngleich solche serverbasierten Systeme im allgemeinen gut funktioniert haben, ergeben sich daraus Probleme, daß auf eine zentralisierte Steuerung des strukturierten Datenspeichers vertraut wird. Beispielsweise hängt der Betrieb des strukturierten Speichersystems von der richtigen Funktionsweise des zentralen Servers ab. Jedesmal, wenn es dem Server nicht gelingt, den richtigen Betrieb aufrechtzuerhalten, beispielsweise bei einem Stromausfall, einem Hardwareausfall oder einem anderen Systemausfall, wird das gesamte strukturierte Speichersystem deaktiviert und verhindert, daß Benutzer auf den Datenspeicher zugreifen. Zusätzlich kann der Serverprozeß durch eine Flut von den einzelnen Netzwerkknoten ausgegebener Client-Dienstanforderungen überlastet werden und das System kann dadurch verlangsamt werden oder zusammenbrechen. Dementsprechend kann das Vertrauen auf eine zentralisierte Steuerung des strukturierten Speichersystems während Zeiträumen einer starken Benutzung zu einem langsamen Betrieb sowie zu Systemausfällen infolge einer Überlastung des zentralen Servers führen.

[0005] Ein zusätzliches Problem, das bei einem Client-Server-Netzwerksystem auftritt, besteht darin, daß es eine statische Betriebsumgebung bereitstellt, die für eine optimale Funktionsweise bei einem bestimmten Niveau der Netzwerkaktivität eingestellt ist. Folglich nutzt das Netzwerk verfügbare Betriebsmittel zum Verbessern der Systemfunktionsweise nicht aus. Insbesondere ist die statische Betriebsumgebung, wenn die Systemaktivität über das erwartete Niveau der Netzwerkaktivität ansteigt oder darunter absinkt, nicht in der Lage, die Zuordnung von Netzwerkbetriebsmitteln dynamisch so zu konfigurieren, daß eine bessere Funktionsweise für das gegenwärtige Aktivitätsniveau bereitgestellt wird.

[0006] Es wurde eine Technologie zum Verbessern der Zuverlässigkeit und der Arbeitsweise dieser zentralisiert strukturierten Speichernetzwerksysteme entwickelt. Diese Technologie war in erster Linie auf die Entwicklung zuverlässiger Datenbanken und Dateisysteme gerichtet, und sie beinhaltete im allgemeinen eines von zwei Verfahren, nämlich (1) das statische Abbilden der Daten auf einen oder mehrere Server oder (2) das Speichern der Daten in einem global gemeinsam benutzten Datenspeicher in der Art einer gemeinsam verwendeten Platte.

[0007] Systeme, bei denen das erste Verfahren verwendet wird, verteilen Abschnitte des strukturierten Speichers permanenter Daten statisch über eine Mehrzahl von Servern. Jeder der Server enthält einen Abschnitt des strukturierten Datenspeichers sowie wahlweise einen zugeordneten Abschnitt einer Verzeichnisstruktur, die die Abschnitte der innerhalb dieses bestimmten Servers gespeicherten Daten beschreibt. Diese Systeme schützen vor einem Datenverlust, indem sie den Datenspeicher statisch über eine Mehrzahl von Servern verteilen, so daß der Ausfall eines Servers zu einem Verlust nur eines Teils der Gesamtdaten führt. Andere Entwicklungen in der Cluster-Datenbanktechnologie sehen das Replizieren von Abschnitten des strukturierten Datenspeichers und das statische Speichern der replizierten Abschnitte über eine Mehrzahl von Servern vor. Dementsprechend gehen diese Systeme beim Schützen vor dem Datenverlust weiter, indem sie innerhalb des strukturierten Speichersystems eine statische Redundanz bereitstellen. Wenngleich die be-

kannte Cluster-Datenbanktechnologie jedoch in der Hinsicht, daß sie vor einem Datenverlust schützt, einen fehlertoleranteren Betrieb bereitstellen kann, beruhen die bekannten Systeme weiterhin auf einer statischen Zuordnung der Daten über verschiedene Server. Weil Daten nicht dynamisch zwischen Servern zugeordnet werden, werden (1) Systembetriebsmittel nicht auf der Grundlage der Systemverwendung zugeordnet, was zu einer Unterbenutzung dieser Betriebsmittel führt, ist (2) die skalierbare Leistungsfähigkeit begrenzt, weil neue Server immer dann bereitgestellt werden müssen, wenn der Datensatz anwächst oder wenn ein bestimmter Server Anforderungen nicht nachkommen kann, die an seinen Abschnitt des Datensatzes gerichtet sind, und (3) erfordert eine solche statische Zuordnung weiterhin, daß wenigstens einer der die Informationen speichernden Server überlebt, um die Daten zu bewahren.

[0008] Systeme, die das zweite Verfahren verwenden, speichern die strukturierten Daten in einem zentralen Datenspeicher, wie bspw. in einer gemeinsam verwendeten Platte. Jeder Knoten in dem System aktualisiert ständig den zentralen Datenspeicher innerhalb seines Abschnitts des strukturierten Speichers. Beispielsweise exportiert in einem Datenbanksystem jeder Knoten Tabellen, die er gegenwärtig verwendet, zum Datenspeicher. Wenngleich dieses Verfahren die Probleme des Lastausgleichs auf den zentralen Datenspeicher überträgt, weist es zwei wesentliche Nachteile auf. Erstens ist der Durchsatz verringert, weil der Zusatzaufwand erhöht ist, der mit dem Gewährleisten der Kohärenz des zentralisierten Datenspeichers verbunden ist. Zweitens ist das Sperren unwirksam, weil ganze Seiten gesperrt werden, wenn ein Knoten auf einen Abschnitt einer Seite zugreift. Folglich können Knoten eine Speicherkonkurrenz erfahren, selbst wenn keine wirklichen Konflikte auftreten.

[0009] Eine weitere Lösung aus dem Stand der Technik wurde in "SERVERLESS NETWORK FILE SYSTEMS" von Anderson, T. E. u. a. (OPERATING SYSTEMS REVIEW (SIGOPS), Band 29, Nr. 5, 1. Dezember 1995, S. 109–126) beschrieben. In diesem Dokument ist ein System vorgeschlagen, bei dem jeder Netzwerkknoten Dateien speichern kann, die für alle anderen Knoten zugänglich sind, so daß jeder Knoten einen Teil des gemeinsam verwendeten Speichers bereitstellen kann. Der Nachteil dieses Systems besteht jedoch darin, daß jeder Knoten, um eine Datei lokalisieren zu können, zwei getrennte Abbildungen unterhalten muß, nämlich eine für die lokal gespeicherten Dateien und eine für die fern gespeicherten Dateien.

[0010] In einer Hinsicht sieht die vorliegende Erfindung ein Verfahren zum Bereitstellen einer verteilten Steuerung eines strukturierten Datenspeichers vor, das die folgenden Schritte aufweist:

Bereitstellen einer Mehrzahl von durch ein Netzwerk miteinander verbundenen Knoten, wobei jeder der Mehrzahl von Knoten einen gemeinsam verwendeten adressierbaren Speicherplatz eines gemeinsam verwendeten Speichersystems geteilt verwendet und (i) eine Schnittstelle zum Zugreifen auf das Netzwerk, (ii) eine lokale flüchtige Speichervorrichtung, die mit dem Knoten gekoppelt ist und einen flüchtigen Speicher bereitstellt, (iii) eine lokale permanente Speichervorrichtung, die mit dem Knoten gekoppelt ist und einen permanenten Speicher bereitstellt, und (iv) ein gemeinsam verwendetes Speicheruntersystem zum Abbilden eines Abschnitts des gemeinsam verwendeten adressierbaren Speicherplatzes in mindestens einen Abschnitt des permanenten und des flüchtigen Speichers, um dadurch einen adressierbaren permanenten und flüchtigen Speicher bereitzustellen, der von jedem der Mehrzahl von Knoten ansteuerbar ist, aufweist, wobei das gemeinsam verwendete Speicheruntersystem (a) einen Verteiler zum Abbilden von Abschnitten des adressierbaren Speicherplatzes über die Mehrzahl von lokalen permanenten und flüchtigen Speichervorrichtungen, um den adressierbaren Speicherplatz über die Mehrzahl von lokalen permanenten und flüchtigen Speichervorrichtungen zu verteilen, und (b) eine Verzeichnisverwaltungseinheit zum Verfolgen der abgebildeten Abschnitte des adressierbaren Speicherplatzes, um Informationen bereitzustellen, die angeben, welche Abschnitte des adressierbaren Speicherplatzes auf welche der lokalen permanenten und flüchtigen Speichervorrichtungen abgebildet sind, aufweist,

Speichern einer Instanz bzw. Ausprägung eines Datensteuerprogramms zum Manipulieren des strukturierten Datenspeichers an jedem Knoten, um mehrere verteilte Ausprägungen des Datensteuerprogramms bereitzustellen,

Verknüpfen jeder Ausprägung des Datensteuerprogramms mit dem gemeinsam verwendeten Speichersystem und

Betreiben jeder Ausprägung des Datensteuerprogramms, um das gemeinsam verwendete Speichersystem als eine Speichervorrichtung zu verwenden, in der der strukturierte Datenspeicher enthalten ist, wobei die Koordinaten des gemeinsam verwendeten Speichersystems auf den strukturierten Datenspeicher zugreifen, um eine verteilte Steuerung des strukturierten Datenspeichers bereitzustellen.

[0011] In einer anderen Hinsicht sieht die vorliegende Erfindung ein Verfahren zum Bereitstellen einer verteilten Steuerung über einen strukturierten Datenspeicher vor, das die folgenden Schritten aufweist:

Bereitstellen einer Mehrzahl von durch ein Netzwerk miteinander verbundener Knoten,

Speichern einer Ausprägung eines Datensteuerprogramms an jedem Knoten, um mehrere verteilte Ausprägungen des Datensteuerprogramms bereitzustellen, wobei das Datensteuerprogramm den Zugriff auf den strukturierten Datenspeicher manipuliert und steuert,

Verknüpfen jeder Ausprägung des Datensteuerprogramms mit einem gemeinsam verwendeten Speichersystem, das einen adressierbaren permanenten

Datenspeicher bereitstellt,
Betreiben jeder Ausprägung des Datensteuerprogramms, um das gemeinsam verwendete Speichersystem als eine Speichervorrichtung zu verwenden, in der der strukturierte Datenspeicher enthalten ist, wobei die Koordinaten des gemeinsam verwendeten Speichersystems auf den strukturierten Datenspeicher zugreifen, um eine verteilte Steuerung des strukturierten Datenspeichers bereitzustellen,
Versehen der strukturierten Datenspeicher mit einem Verzeichnis, und
Betreiben des gemeinsam verwendeten Speichersystems, um das Verzeichnis innerhalb eines gemeinsam verwendeten Speicherplatzes zu erhalten.

[0012] Bevorzugte Ausführungsformen der Erfindung sehen verbesserte Speichersysteme zum Unterhalten eines strukturierten Datenspeichers, strukturierte Speichersysteme, die zuverlässiger sind, einen fehlertoleranteren Betrieb bereitstellen und die Fähigkeit aufweisen, Daten ansprechend auf Netzwerkaktivitätsniveaus und Zugriffsmuster dynamisch zu bewegen, um die Funktionsweise zu optimieren und Knotenzugriffszeiten zu minimieren, strukturierte Speichersysteme, die eine verteilte Steuerung über einen strukturierten Speicher permanenter Daten bereitstellen, wobei die Daten beispielsweise Computerdateien, Datenbank-Datensätze und Informationen oder Web-Seiten einschließen können, vor, und sie sehen eine verteilte Steuerung einer Mehrzahl verschiedener Typen strukturierter Speichersysteme, wie Dateisysteme, Datenbanksysteme und Systeme, die Web-Seiten speichern, gemeinsam verwenden und sie zu anfordernden Knoten und/oder anfordernden Netzwerken übermitteln, vor.

[0013] Ausführungsformen der Erfindung können als strukturierte Speichersysteme und verwandte Verfahren, bei denen ein global adressierbares unstrukturiertes Speichersystem verwendet wird, um einen strukturierten Speicher permanenter Daten innerhalb eines gemeinsam verwendeten Speicherplatzes zu unterhalten, verstanden werden. Es kann ein gemeinsam verwendetes Speichersystem, wie bspw. ein verteiltes gemeinsam verwendetes Speichersystem (DSM), verwendet werden, das den Datenspeicher über einige oder alle mit einem Netzwerk verbundenen Speichervorrichtungen verteilt. Speichervorrichtungen, die mit dem Netzwerk verbunden werden können, umfassen Festplatten-Laufwerke, Bandlaufwerke, Diskettenlaufwerke, CD-ROM-Laufwerke, optische Plattenlaufwerke, Direktzugriffsspeicher-Chips oder Nurlesespeicher-Chips.

[0014] Das strukturierte Speichersystem kann ein Computerprogramm sein, das sich mit einem DSM austauscht, um das DSM als eine Speichervorrichtung zu betreiben, die einen permanenten Datenspeicher bereitstellt. Das Steuerprogramm des strukturierten Speichersystems kann dem DSM vorschreiben, Datei- und Verzeichnisdaten in den gemeinsam verwendeten Speicherplatz abzubilden. Das DSM kann Funktionen zum gemeinsamen Verwenden und kohärenten Replizieren von Daten aufweisen. Bei einer Ausführungsform stellt das DSM dem Datensteuerprogramm Speichervorrichtungsdienste bereit. Diese Dienste können Lese-, Schreib-, Zuordnungs-, Räumungs- oder andere ähnliche oder zusätzliche Dienste einschließen, die geeignet sind, eine Steuerung niedriger Ebene einer Speichervorrichtung bereitzustellen. Das Datensteuerprogramm verwendet diese DSM-Dienste zum Zuordnen von Abschnitten des gemeinsam verwendeten Speicherplatzes und zum Zugreifen auf diese, um einen strukturierten Speicher permanenter Daten zu erzeugen und zu manipulieren.

[0015] Bei einer Ausführungsform sieht die Erfindung ein Verfahren und ein verwandtes System zum Bereitstellen einer verteilten Steuerung des strukturierten Datenspeichers vor. Das Verfahren umfaßt das Bereitstellen einer Mehrzahl durch ein Netzwerk miteinander verbundener Knoten und das Speichern einer Ausprägung eines Datensteuerprogramms zum Manipulieren des strukturierten Datenspeichers zum Bereitstellen mehrerer verteilter Ausprägungen des Datensteuerprogramms auf jedem Knoten. Das Verfahren umfaßt auch das Verknüpfen jeder Ausprägung des Datensteuerprogramms mit einem gemeinsam verwendeten Speichersystem, das einen adressierbaren permanenten Datenspeicher bereitstellt und das Betreiben jeder Ausprägung des Datensteuerprogramms, um das gemeinsam verwendete Speichersystem als eine Speichervorrichtung zu verwenden, in der der strukturierte Datenspeicher enthalten ist, wobei die Koordinaten des gemeinsam verwendeten Speichersystems auf den strukturierten Datenspeicher zugreifen, um eine verteilte Steuerung des strukturierten Datenspeichers bereitzustellen.

[0016] Ausführungsformen der Erfindung umfassen das Verknüpfen aller Ausprägungen des Datensteuerprogramms mit einem DSM, das eine verteilte Speicherung über die miteinander verbundenen Knoten bereitstellt und das einen permanenten Speicher für die Daten bereitstellt. Der Verknüpfungsschritt kann weiterhin einschließen, daß dem Datensteuerprogramm vorgeschrieben wird, einen im strukturierten Datenspeicher zu speichernden Datenstrom bereitzustellen und das gemeinsam verwendete Speichersystem als eine Einzelknoten-Speichervorrichtung zu betreiben.

[0017] Andere Ausführungsformen der Erfindung umfassen das Betreiben des gemeinsam verwendeten Speichersystems, so daß gespeicherte Daten kohärent repliziert werden, um einen redundanten Datenspeicher bereitzustellen, und das Speichern der kohärent replizierten Daten innerhalb verschiedener Speichervorrichtungen des Netzwerks, um einen fehlertoleranten Betrieb bereitzustellen. Weiterhin sind das Koordinieren des gemeinsamen Zugriffs auf Daten innerhalb des strukturierten Speichers durch innerhalb eines gemeinsam verwendeten Speicherplatzes gespeicherten Sperrobjekte und das Erzeugen einer Sperrobjekt-Datenstruktur mit Informatio-

nen, die einen Sperrstatus darstellen, auf Abschnitten des gemeinsam verwendeten Speicherplatzes und das Speichern des Sperrobjekts innerhalb des gemeinsam verwendeten Speicherplatzes zum Bereitstellen einer Sperre des gemeinsam verwendeten Systems enthalten. Es können Objekte gesperrt werden, indem dem gemeinsamen Speicher vorgeschrieben wird, Sperren für Abschnitte des gemeinsam verwendeten Speicherplatzes zu erzeugen. Weiterhin kann das Datensteuerprogramm Daten komprimieren, die in dem strukturierten Datenspeicher zu speichern sind.

[0018] Andere Ausführungsformen der Erfindung umfassen Anordnungen, bei denen der strukturierte Datenspeicher ein Dateisystem, ein Datenbanksystem, ein Web-Seitensystem oder allgemein ein beliebiges Objekte speicherndes, abrufendes, manipulierendes und zuführendes System aufweist. Für die Ausführungsform des Dateisystems weist das Datensteuerprogramm ein Dateisteuerprogramm zum Manipulieren des Dateisystems auf, wodurch das gemeinsam verwendete Speichersystem den Zugriff auf das Dateisystem steuert, um ein gemeinsam verwendetes Dateisystem bereitzustellen. Für die Ausführungsform des Datenbanksystems weist das Datensteuerprogramm ein Datenbank-Steuerprogramm zum Manipulieren des Datenbanksystems auf, wodurch das gemeinsam verwendete Speichersystem den Zugriff auf das Datenbanksystem steuert, um ein gemeinsam verwendetes Datenbanksystem bereitzustellen. Für die Ausführungsform des Web-Seitensystems weist das Datensteuerprogramm ein Web-Seitensteuerprogramm zum Manipulieren des Web-Seitensystems auf, wodurch das gemeinsam verwendete Speichersystem den Zugriff auf das Web-Seitensystem steuert, um ein gemeinsam verwendetes Web-Seitensystem bereitzustellen. Für beliebige dieser speziellen Ausführungsformen verwendet das gemeinsam verwendete System ein Verzeichnis und betreibt das gemeinsam verwendete Speichersystem, um das Verzeichnis innerhalb des gemeinsam verwendeten Speicherplatzes aufrechtzuerhalten, und das Verzeichnis ist als eine Mehrzahl innerhalb des gemeinsam verwendeten Speicherplatzes gespeicherter Sätze organisiert. Weiterhin wird für ein innerhalb des gemeinsam verwendeten Systems gespeichertes Objekt (beispielsweise eine Datei, ein Datenbank-Datensatz, eine Web-Seite usw.) ein Deskriptor erzeugt, der einen Speicher für einen Bezeichner aufweist, der einen Abschnitt eines gemeinsam verwendeten Speicherplatzes darstellt, und es können benachbarte Abschnitte des gemeinsam verwendeten Speicherplatzes zugeordnet werden, die jeweils durch einen jeweiligen Bezeichner dargestellt sind, um reduzierte Buchhaltungsinformationen für die jeweilige Datei bereitzustellen und den Zugriff auf den physikalischen Speicher für die Datei zu optimieren.

[0019] Einige Ausführungsformen der Erfindung werden nun nur als Beispiel mit Bezug auf die anliegende Zeichnung beschrieben. In der Zeichnung bezeichnen gleiche Bezugszahlen im allgemeinen überall in den verschiedenen Darstellungen die gleichen Teile. Weiterhin sind die Zeichnungen nicht notwendigerweise maßstabsgerecht, wobei der Nachdruck im allgemeinen vielmehr auf das Erläutern der Grundgedanken der Erfindung gelegt wird.

[0020] **Fig.** 1 zeigt ein schematisches Blockdiagramm eines verteilten, adressierbaren, strukturierten gemeinsam verwendeten Datenspeichersystems gemäß der Erfindung.

[0021] **Fig.** 2 zeigt ein Diagramm einer möglichen Ausführungsform des Systems aus **Fig.** 1, nämlich ein verteiltes, adressierbares gemeinsam verwendetes Speicher-Dateisystem, das einen Speicher für Computerdateien, wie bspw. Source-Code-Dateien, Textverarbeitungsdokument-Dateien usw., bereitstellt.

[0022] **Fig.** 3 zeigt eine graphische Darstellung der Organisation von Verzeichniseinträgen und zugeordneten Datei-Deskriptoren (auch als "Inoden" bekannt), die zur Verwendung mit dem Dateisystem aus **Fig.** 2 geeignet ist.

[0023] **Fig.** 4 zeigt ein Diagramm eines zur Verwendung mit dem Dateisystem aus **Fig.** 2 geeigneten Inoden.

[0024] **Fig.** 5 zeigt ein verteiltes gemeinsam verwendetes Speicher-Computernetzwerk.

[0025] **Fig.** 6 zeigt ein Funktionsblockdiagramm, in dem ein verteiltes gemeinsam verwendetes Speicher-Computernetzwerk des in **Fig.** 5 dargestellten Typs in weiteren Einzelheiten dargestellt ist.

[0026] **Fig.** 7 zeigt in weiteren Einzelheiten ein gemeinsam verwendetes Speicheruntersystem, das zum Einsatz mit dem in 6 dargestellten Netzwerk geeignet ist.

[0027] **Fig.** 8 zeigt ein Funktionsblockdiagramm eines gemeinsam verwendeten Speicheruntersystems gemäß der Erfindung.

[0028] **Fig.** 9 zeigt eine Verzeichnisseite, die durch ein gemeinsam verwendetes Speicheruntersystem des in **Fig.** 8 dargestellten Typs bereitgestellt werden kann.

[0029] **Fig.** 10 zeigt ein Verzeichnis, das innerhalb eines gemeinsam verwendeten Speichers verteilt werden kann und aus Verzeichnisseiten des in **Fig.** 9 dargestellten Typs besteht.

[0030] **Fig.** 11 zeigt in Form eines Funktionsblockdiagramms ein System, bei dem ein Verzeichnis gemäß **Fig.** 10 zum Verfolgen von Abschnitten eines verteilten gemeinsam verwendeten Speichers verwendet wird.

Beschreibung

[0031] Ein Netzwerksystem **10** gemäß der Erfindung weist eine Mehrzahl von Netzwerkknoten auf, die auf einen Speicherplatz zugreifen, in dem ein strukturierter Datenspeicher, wie bspw. ein strukturiertes Dateisystem oder eine Datenbank, gespei-

chert ist. Jeder der Knoten weist mindestens ein Datensteuerprogramm auf, das auf den strukturierten Datenspeicher zugreift und diesen verwaltet. Der strukturierte Datenspeicher kann in einem adressierbaren gemeinsam verwendeten Speicher gespeichert sein, oder der strukturierte Speicher kann in traditionellerer Weise gespeichert sein. Beispielsweise kann jeder Knoten dafür verantwortlich sein, ein bestimmtes Element oder bestimmte Elemente des strukturierten Datenspeichers zu speichern. Bei einer solchen Ausführungsform kann das Datensteuerprogramm unter Verwendung eines global eindeutigen Bezeichners auf einen gewünschten Abschnitt des strukturierten Speichers zugreifen. Das zugrundeliegende System übersetzt den Bezeichner in einen oder mehrere Befehle unter Einschluß von Netzwerkübertragungsbefehlen zum Zugreifen auf die gewünschten Daten. Bei einer weiteren Ausführungsform wird der strukturierte Datenspeicher in einem adressierbaren gemeinsam verwendeten Speicherplatz gespeichert, wodurch ermöglicht wird, daß die Knoten unter Verwendung von Standard-Speicherzugriffsbefehlen transparent auf Abschnitte des strukturierten Speichers zugreifen.

[0032] Das System **10** kann ein Dateisystem, ein Datenbanksystem, ein Web-Server, ein Objektquellensystem oder ein anderes strukturiertes Speichersystem sein, das einen organisierten Datensatz enthält. Der Begriff "Web-Server" soll hier einen beliebigen Prozessor bezeichnen, der Datenobjekte (wie bspw. Active-X-Objekten), Anwendungen (wie bspw. JAVA-Applets) oder Dateien (wie bspw. HTML-Dateien) über Web-Protokolle (beispielsweise HTTP oder FTP) zu einem Requestor überträgt. Bei einer offenbarten Ausführungsform ist das System **10** ein Dateisystem, das verschiedene Computerdateien enthält. Dies ist jedoch lediglich eine Ausführungsform der Erfindung, die Erläuterungszwecken dient. Die Erfindung kann verwendet werden, um beliebige einer Mehrzahl strukturierter Speichersysteme (beispielsweise ein Datenbanksystem, ein Web-Seitensystem, ein Intranet usw.) bereitzustellen. Die Erfindung soll nicht auf das Dateisystem oder andere spezielle hier beschriebene Ausführungsformen beschränkt sein.

[0033] Mit Bezug auf **Fig.** 1 sei bemerkt, daß ein Netzwerksystem **10** gemäß der Erfindung eine Mehrzahl von Netzwerkknoten **12a**–**12d** und einen adressierbaren gemeinsam verwendeten Speicherplatz **20** aufweist, wobei der gemeinsam verwendete Speicherplatz **20** einen Abschnitt **22** zum Speichern eines strukturierten Datenspeichers **28** aufweist. Jeder der Knoten **12a**–**12d** kann mehrere Unterelemente aufweisen. Beispielsweise weist der Knoten **12a** einen Prozessor **30a**, ein Datensteuerprogramm **32a** und ein gemeinsam verwendetes Speicheruntersystem **34a** auf. Bei der offenbarten Ausführungsform weisen zwei der Knoten **12a** und **12c** Bildschirme auf, die Anzeigen **40** und **42** bereitstellen, die den strukturierten Datenspeicher **28** innerhalb des adressierbaren gemeinsam verwendeten Speicherplatzes **20** graphisch darstellen. Der adressierbare gemeinsam verwendete Speicherplatz **20** verbindet alle Netzwerkknoten **12a**–**12d** miteinander und bietet jedem der Knoten **12a**–**12d** einen Zugriff auf den im adressierbaren gemeinsam verwendeten Speicherplatz **20** enthaltenen strukturierten Datenspeicher **28**.

[0034] Ein erfindungsgemäßes System **10** kann unter anderem jeden Netzwerkknoten **12a**–**12d** mit einer geteilten Steuerung des strukturierten Datenspeichers **28** versehen, und das System **10** kann daher die Steuerung des Datenspeichers über die Knoten des Netzwerks verteilen. Hierzu weist jeder Knoten des Systems **10** in der Art des Knotens **12a** ein Datensteuerprogramm **32a** auf, das mit einem gemeinsam verwendeten Speicheruntersystem **34a** verknüpft ist. Das Datensteuerprogramm **32a** kann als ein strukturiertes Speichersystem in der Art eines Dateisystems arbeiten, das dafür eingerichtet ist, einen strukturierten Datenspeicher zu erhalten und das gemeinsam verwendete Speichersystem als eine adressierbare Speichervorrichtung zu verwenden, die einen strukturierten Datenspeicher speichern kann. In Richtung des Datensteuerprogramms **32a** kann das gemeinsam verwendete Speicheruntersystem **34a** auf Daten innerhalb des adressierbaren gemeinsam verwendeten Speicherplatzes **20** zugreifen und diese speichern. Diese zusammenwirkenden Elemente bilden ein strukturiertes Speichersystem, das eine verteilte Architektur aufweist und dadurch eine größere Fehlertoleranz, Zuverlässigkeit und Flexibilität erreicht als dies bei bekannten strukturierten Speichersystemen der Fall ist, die auf einer zentralisierten Steuerung und zentralisierten Servern beruhen. Dementsprechend kann die Erfindung Computernetzwerke mit verteilt gesteuerten und leicht skalierbaren Dateisystemen, Datenbanksystemen, Web-Seitensystemen, Objektquellen, Daten-Cache-Systemen oder anderen strukturierten Speichersystemen bereitstellen.

[0035] Weiterhin mit Bezug auf **Fig.** 1 sei bemerkt, daß das erfindungsgemäße System **10** innerhalb des adressierbaren gemeinsam verwendeten Speicherplatzes **20** einen strukturierten Datenspeicher **28** betreibt. Jeder der Knoten **12a**–**12d** kann über die gemeinsam verwendeten Speicheruntersysteme **34a**–**34d** auf den adressierbaren gemeinsam verwendeten Speicherplatz **20** zugreifen. Jedes der gemeinsam verwendeten Speicheruntersysteme **34a**–**34d** bietet seinem Knoten einen Zugriff auf den adressierbaren gemeinsam verwendeten Speicherplatz **20**. Die gemeinsam verwendeten Speicheruntersysteme **34a**–**34d** koordinieren jede der Speicherzugriffsoperationen des jeweiligen Knotens, um einen Zugriff auf die gewünschten Daten bereitzustellen und innerhalb des adressierbaren gemeinsam verwendeten Speicherplatzes **20** die Datenkohärenz zu erhalten. Dies ermöglicht es, daß die miteinander verbundenen Knoten **12a**–**12d** den adressierbaren gemeinsam verwendeten Speicherplatz **20** als einen Platz zum Speichern und Abrufen von Daten verwen-

den. Wenigstens ein Abschnitt des adressierbaren gemeinsam verwendeten Speicherplatzes **20** wird durch ein physikalisches Speichersystem unterstützt, das einen permanenten Datenspeicher bereitstellt. Beispielsweise kann ein Abschnitt des adressierbaren gemeinsam verwendeten Speicherplatzes **20** einem oder mehreren Festplattenlaufwerken, die sich auf dem Netzwerk befinden oder einem oder mehreren der Netzwerkknoten **12a**–**12d** als lokaler Festplattenspeicher für diese bestimmten Knoten zugeordnet sind, zugewiesen werden oder darauf abgebildet werden. Demgemäß zeigt **Fig.** 1 das erfindungsgemäße System mit gemeinsam verwenden Speicheruntersystemen, die den Netzwerkknoten einen Zugriff auf einen adressierbaren gemeinsam verwendeten Speicherplatz bieten, wobei wenigstens ein Teil dieses Raums wenigstens einem Teil von einer oder mehreren permanenten Speichervorrichtungen (beispielsweise Festplatten) zugewiesen ist, um zu ermöglichen, daß die Knoten Daten adressierbar in einer oder mehreren permanenten Speichervorrichtungen speichern oder aus diesen abrufen. Eine bevorzugte Ausführungsform eines solchen adressierbaren gemeinsam verwendeten Speicherplatzes ist in der am 22. November 1996 eingereichten gemeinhin anerkannten US-Patentanmeldung mit der laufenden Nummer 08/754 481 beschrieben.

[0036] Daher besteht eine Erkenntnis der vorliegenden Erfindung darin, daß jeder der Knoten **12a**–**12d** sein jeweiliges gemeinsam verwendetes Speicheruntersystem als eine Speichervorrichtung verwenden kann, die einen permanenten Datenspeicher bereitstellt.

[0037] Jedes der Datensteuerprogramme **32a**–**32d** ist ein Softwaremodul, das in einer Weise mit dem jeweiligen gemeinsam verwendeten Speicheruntersystem **34a**–**34d** gekoppelt ist, das in ähnlicher Weise wie eine Schnittstelle zwischen einem herkömmlichen Datenspeicherprogramm und einer lokalen Speichervorrichtung arbeitet. Beispielsweise kann das Datensteuerprogramm **32a** Daten zu dem gemeinsam verwendeten Speicheruntersystem **34a** leiten und Daten von diesem aufnehmen. Weil die gemeinsam verwendeten Speicheruntersysteme die Speicherzugriffe auf den adressierbaren gemeinsam verwendeten Speicherplatz **20** koordinieren, wird jedes der Datensteuerprogramme davon befreit, daß es seine Aktivitäten mit den anderen Datensteuerprogrammen auf dem Netzwerk verwalten und koordinieren muß oder daß es seine Aktivitäten mit einem oder mehreren zentralen Servern verwalten und koordinieren muß. Dementsprechend kann jedes der Datensteuerprogramme **32a**–**32d** eine Peer-Ausgestaltung (also eine Instanz bzw. Ausprägung) sein, die an einem anderen der Netzwerkknoten **12a**–**12d** vorhanden ist und das jeweilige gemeinsam verwendete Speicheruntersystem **34a**– **34d** als eine lokale Speichervorrichtung in der Art einer lokalen Festplatte behandeln kann.

[0038] Eines oder mehrere der Datensteuerprogramme **32a**–**32d** kann eine graphische Benutzerschnittstelle **42** bereitstellen, die den innerhalb des adressierbaren gemeinsam verwendeten Speicherplatzes **20** enthaltenen strukturierten Datenspeicher **28** graphisch darstellt. Die graphische Benutzerschnittstelle **42** ermöglicht es einem Benutzer an einem Knoten, beispielsweise an einem Knoten **12a**, Datenobjekte graphisch innerhalb des strukturierten Datenspeichers **28** einzufügen. Hierzu kann das Datensteuerprogramm **32a** einen Befehlssatz erzeugen, der dem gemeinsam verwendeten Speicheruntersystem **34a** einen Datenstrom zuführt, und das gemeinsam verwendete Speicheruntersystem **34a** verwendet den Datenstrom zum Speichern eines Objekts innerhalb des strukturierten Datenspeichers **28**. In ähnlicher Weise können die anderen gemeinsam verwendeten Speicheruntersysteme **34b**–**34d** ihren jeweiligen Knoten Informationen zuführen, die diese Änderung an dem strukturierten Datenspeicher **28** angeben. Dementsprechend reflektiert dieser Knoten (der eine graphische Benutzerschnittstelle **40** aufweist), wie in **Fig.** 1 nur der Einfachheit halber für den Knoten **12c** dargestellt ist, die vom Datensteuerprogramm **32a** des Knotens **12a** bewirkte Änderung am strukturierten Datenspeicher **28**. Insbesondere kann die graphische Benutzerschnittstelle **40** des Knotens **12c** einem Benutzer zeigen, daß ein Objekt in den strukturierten Datenspeicher **28** gegeben wird. Beispielsweise enthält der adressierbare gemeinsam verwendete Speicherplatz **20** auch die Datenobjekte **50a**–**50c**, die in den strukturierten Datenspeicher **28** gegeben werden können, um Teil dieses strukturierten Datenspeichers zu werden. Wie dargestellt ist, kann ein Systembenutzer am Knoten **12a** vorschreiben, daß ein Objekt **50a** an einer festgelegten Stelle innerhalb des Datenspeichers **28** eingefügt wird. Das Datensteuerprogramm **32a** schreibt dem gemeinsam verwendeten Speicheruntersystem **34a** dann vor, das Objekt **50a** an der geeigneten Stelle in den Datenspeicher **28** zu geben. Weiterhin erfaßt das gemeinsam verwendete Speicheruntersystem **34c** am Knoten **12c** die Änderung innerhalb des Datenspeichers **28** und reflektiert diese Änderung innerhalb der graphischen Benutzerschnittstelle **40**.

[0039] Mit Bezug auf **Fig.** 2 sei bemerkt, daß ein strukturiertes Dateisystem **60** eine spezielle Ausführungsform gemäß der Erfindung ist, bei der die Eigenschaften des adressierbaren gemeinsam verwendeten Speicherplatzes **20** verwendet werden, um etwas zu implementieren, was für alle Netzwerkknoten wie ein kohärentes, einzelnes Dateisystem aussieht, während es tatsächlich alle mit dem adressierbaren gemeinsam verwendeten Speicherplatz **20** gekoppelten Netzwerkknoten umfaßt.

[0040] Das Dateisystem **60** aus **Fig.** 2 unterscheidet sich von bekannten physikalischen und verteilten Dateisystemen in mehrfacher Hinsicht. Im Gegensatz zu bekannten physikalischen Dateisystemen, die eine Dateiorganisation auf Plattenblöcke abbilden, verwaltet das Dateisystem **60** gemäß der Erfin-

dung die Abbildung einer Verzeichnis- und Dateistruktur auf ein verteiltes, adressierbares gemeinsam verwendetes Speichersystem **20**, bei dem wenigstens ein Teil seines adressierbaren Platzes auf wenigstens einen Teil von einer oder mehreren permanenten Speichervorrichtungen (beispielsweise Festplatten) auf dem Netzwerk abgebildet ist oder diesem zugewiesen ist. Anders als bekannte verteilte Dateisysteme verwendet das Dateisystem **60** gemäß der Erfindung Peer-Knoten, die jeweils eine Ausgestaltung oder Ausprägung desselben Datensteuerprogramms aufweisen. Weiterhin führt das Dateisystem **60** gemäß der Erfindung, anders als bekannte Dateisysteme, generell folgendes aus: Es erhält die Datenkohärenz zwischen Netzwerkknoten, es repliziert Daten automatisch zum Erhalten einer Redundanz und Fehlertoleranz, es übermittelt Daten automatisch und dynamisch, um einer sich ändernden Netzwerkverwendung und sich ändernden Verkehrsmustern Rechnung zu tragen, und es bietet eine Vielzahl anderer Fortschritte und Vorteile, von denen einige in der am 22. November 1996 eingereichten anhängigen US-Patentanmeldung mit der laufenden Nummer 08/754 481 beschrieben sind.

[0041] Weiterhin mit Bezug auf **Fig.** 2 sei bemerkt, daß das Dateisystem **60** teilweise innerhalb des adressierbaren gemeinsam verwendeten Speicherplatzes **20** vorhanden ist und einen strukturierten Datenspeicher **62**, eine Über-Wurzel **64**, Dateisätze **66**–**74**, einen Verzeichniseintrag **80** und eine Datei oder ein Dokument **82** aufweist. Es sind zwei Netzwerkknoten **84** und **86** dargestellt, die über die logischen Laufwerke **90** und **94** auf den adressierbaren gemeinsam verwendeten Speicherplatz **20** zugreifen (in der vorstehend mit Bezug auf **Fig.** 1 beschriebenen Weise). Anwendungsprogramme **92** und **96**, die auf den Knoten ausgeführt werden, tauschen sich mit den Datensteuerprogrammen (in **Fig.** 2 nicht dargestellt, jedoch in **Fig.** 1 als **32a**–**32d** dargestellt) aus und bewirken, daß die Datensteuerprogramme in den Knoten auf die logischen Laufwerke **90** und **94** zugreifen. Bei der offenbarten Ausführungsform sind die logischen Laufwerke DOS-Vorrichtungen, die sich über installierbare Dateisystem-Treiber, die dem Dateisystem **60** zugeordnet sind, mit den Verzeichnissen des Dateisatzes verbinden.

[0042] Das Dateisystem **60** unterstützt ein globales Dateisystem je adressierbarem gemeinsam verwendetem Speicherplatz **20**, das von allen Netzwerkknoten gemeinsam verwendet wird. Dieses globale Dateisystem ist in ein oder mehrere unabhängige Dateisammlungen organisiert, die als Dateisätze **66**–**74** dargestellt sind. Ein Dateisatz kann als einer traditionellen Dateisystempartition logisch äquivalent angesehen werden. Er ist eine Ansammlung von Dateien, die hierarchisch als eine Verzeichnisbaumstruktur organisiert sind, deren Stamm in einem Wurzelverzeichnis liegt. Die nicht verzweigten Knoten in dem Baum sind die Verzeichnisse **80**, und die Zweige in dem Baum sind reguläre Dateien **82** oder leere Verzeichnisse. Unterverzeichnisbäume innerhalb eines Dateisatzes können durch Verknüpfen einer Datei mit mehreren Verzeichnissen überlappen.

[0043] Ein Vorteil des Aufbrechens des Dateisystems **60** in Dateisätze **66**–**74** besteht darin, daß den Benutzern des Systems **60** dadurch eine flexiblere Dateisystemverwaltung bereitgestellt wird. Wenn das Dateisystem **60** zu einer sehr hohen Größe anwächst (beispielsweise Hunderte von Knoten mit Tausenden von Gigabits an Speicherplatz), ist es erwünscht, die Dateien zu Gruppen von Verwaltungseinheiten zu organisieren, so daß Verwaltungsaktionen unabhängig auf einzelne Gruppen angewendet werden können, ohne die Funktionsweise der anderen zu beeinflussen.

[0044] Die Dateisätze in dem adressierbaren gemeinsam verwendeten Speicherplatz **20** sind in einer gemeinsamen Struktur beschrieben und spezifiziert, deren Wurzel **64** den Anfangspunkt zum Lokalisieren der Dateisätze in dem adressierbaren gemeinsam verwendeten Speicherplatz **20** bildet. Die Wurzel **64** kann an einer statischen und wohlbekannten Speicherstelle in dem adressierbaren gemeinsam verwendeten Speicherplatz **20** gespeichert werden, und auf sie kann über eine Programmschnittstelle des verteilten gemeinsam verwendeten Speichersystems zugegriffen werden. Wenn ein Knoten zum ersten Mal auf einen Dateisatz zugreift, sieht er zunächst in der Wurzel **64** nach, um den Bezeichner zu bestimmen, der dem Dateisatz zugeordnet ist, beispielsweise die Adresse des gemeinsam verwendeten Speichers, die zum Zugreifen auf den Dateisatz verwendet wird. Sobald er den Bezeichner bestimmt hat, kann der Knoten auf das Wurzelverzeichnis des Dateisatzes zugreifen. Von dem Wurzelverzeichnis kann er den gesamten Verzeichnisbaum des Dateisatzes durchlaufen, um die gewünschte Datei zu lokalisieren. Von dem Dateisystem **60** verwendete Dateisätze werden nachstehend in näheren Einzelheiten unter der Überschrift "Dateisatz" beschrieben.

[0045] Mit Bezug auf **Fig.** 3 sei bemerkt, daß bei der offenbarten Ausführungsform des Dateisystems **60** gemäß der Erfindung auf ein Verzeichnis **126** (wie bspw. das Verzeichnis **80** aus **Fig.** 2) zugegriffen wird, wobei bei einem Verzeichnis-Inode oder Deskriptor **128** begonnen wird, der eine Adresse enthält, die auf einen Verzeichniseintragsstrom-Deskriptor **130** verweist. Dieser Deskriptor **130** ist ein Zeiger auf einen Datenblock, der Verzeichniseinträge für die Dateien Datei 1 bis Datei 3 enthält. Der Verzeichniseintrag für die Datei 1 weist eine Anzahl von Einträgen auf, wobei einer der Einträge eine Zeichenkette ist, die den Namen der Datei enthält, und ein anderer Eintrag die Adresse der Inoden und Strom-Deskriptoren **132** ist. Die Strom-Deskriptoren für die Datei 1 werden verwendet, um die verschiedenen 4-Kilobyte-Seiten in dem adressierbaren gemeinsam verwendeten Speicherplatz **20**, die die Datei 1 bilden, zu lokalisieren und abzurufen. Andere Dateien werden in gleicher Weise aus dem adressierbaren gemein-

sam verwendeten Speicherplatz **20** abgerufen und aufgebaut. Die von dem Dateisystem **60** verwendeten Verzeichnisse werden nachstehend unter der Überschrift "Verzeichnis" in näheren Einzelheiten beschrieben.

[0046] Bei der in **Fig.** 4 offenbarten Ausführungsform des Dateisystems **60** wird eine Datei **98** (wie bspw. die Datei **82** aus **Fig.** 2) durch eine oder mehrere gemeinsam verwendete Datenseiten **100**, **102**, **104**, **106** und **108** in dem adressierbaren gemeinsam verwendeten Speicherplatz **20** dargestellt. Jede Datei **98** weist einen Datei-Inoden oder Deskriptor **110** auf, der verschiedene Dateiattribute **112** enthält. Der Datei-Deskriptor **110** enthält eine Adresse, die auf einen Datenstrom-Deskriptor **114** verweist, und der Datenstrom selbst weist eine oder mehrere Adressen **116**, **118**, **120**, **122** und **124** auf, die auf bestimmte Seiten in dem identifizierbaren gemeinsam verwendeten Speicherplatz **20** verweisen. Bei der offenbarten Ausführungsform ist eine Seite die atomare Einheit in dem adressierbaren gemeinsam verwendeten Speicherplatz **20**, und sie enthält bis zu 4 Kilobytes an Daten. Selbst wenn die ganzen 4 Kilobytes nicht erforderlich sind, wird eine ganze Seite verwendet. Dies ist durch die Seite **108** dargestellt, die nur etwa 2 Kilobytes an Daten enthält. Die von dem Dateisystem **60** verwendeten Dateien werden nachstehend in näheren Einzelheiten unter der Überschrift "Dateien" beschrieben.

DATEISATZ

[0047] Die Dateisätze sind die grundlegende Einheit für das Dateisystem **60**. Jeder Dateisatz ist mit einem Namen identifiziert, der bis zu 255 Zeichen aufweist. Das Dateisystem **60** exportiert einen Satz von Operationen auf der Dateisatzebene, die es einem Administrator ermöglichen, die Dateisätze durch Aktionen des folgenden Typs zu verwalten.

Dateisatz-Erzeugung

[0048] Diese Operation erzeugt einen neuen Dateisatz. Der Dateisatz wird zunächst mit einer Datei, dem leeren Wurzelverzeichnis, erzeugt. Ein Standard-Dateisatz wird bei der Initialisierung des adressierbaren gemeinsam verwendeten Speicherplatzes **20** automatisch erzeugt.

Dateisatz-Löschung

[0049] Diese Operation löscht einen Dateisatz. Alle Dateien in dem Dateisatz werden entfernt, und der gesamte gemeinsam verwendete Speicherplatz, der den Dateien in dem Dateisatz zugeordnet ist, wird verworfen, und der zugrundeliegende physikalische Speicher wird für ein neues Speichern befreit. Das Dateisystem **60** ermöglicht nur das Löschen eines Dateisatzes, bis es keine offenen Handles für den Dateidatenstrom in dem Dateisatz gibt. Um einen Dateisatz für das Löschen bereit zu machen, muß der Dateisatz "heruntergefahren" werden, indem er offline geschaltet wird.

Dateisatz-Spezifikation

[0050] Diese Operation spezifiziert einen bestimmten Dateisatz oder alle Dateisätze in dem adressierbaren gemeinsam verwendeten Speicherplatz **20**.

Dateisatz-Steuerung

[0051] Diese Operation führt Steuerroutinen auf der Dateisatz-Ebene in der Art des Festlegens von Dateisatz-Attributen aus.

Aufbauen der Exportsteuerung

[0052] Verzeichnisse werden an lokale Vorrichtungen gebunden, also unter Verwendung von Parametern "aufgebaut", die in der Windows-NT-Registry oder einem anderen ähnlichen zentralen Speicherbereich für solche Informationen gespeichert sind. Wenn es zum ersten Mal laufen gelassen wird, greift das Datensteuerprogramm auf den zentralen Speicher zu und bestimmt, welche Dateisätze aufgebaut werden sollten. Das Datensteuerprogramm erzeugt ein Dateiobjekt, das jeden von den Einträgen im zentralen Speicher identifizierten Dateisatz darstellt. Bei einigen Ausführungsformen kann eine API bereitgestellt werden, die es ermöglicht, daß das Datensteuerprogramm Dateisätze durch Ausführen geeigneter API-Aufrufe dynamisch aufbaut und abbaut.

[0053] Die Benutzer des Dateisystems **60** sind sich nicht des "logischen Datenträgers" bewußt, sondern betrachten vielmehr jeden Dateisatz als einen Datenträger (oder eine Partition im Sinne eines traditionellen physikalischen Dateisystems). GetVolumeInformation aus Win32 wird verwendet, um Informationen über den Dateisatz zu erhalten (genauer gesagt über die logische Vorrichtung, mit der der Dateisatz verbunden ist). Weil alle Dateisätze den gleichen Pool des Speichers im adressierbaren gemeinsam verwendeten Speicherplatz **20** gemeinsam verwenden, ist die dem Benutzer für jeden Dateisatz zurückgegebene Gesamtdatenträgergröße die aktuelle zusammengenommene Speicherkapazität in dem adressierbaren gemeinsam verwendeten Speicherplatz **20**. Das gleiche Verfahren wird für die Informationen zum gesamten freien Platz verwendet, und der Gesamtwert des adressierbaren gemeinsam verwendeten Speicherplatzes **20** wird für jeden Dateisatz zurückgegeben.

VERZEICHNIS

[0054] Das Durchsuchen von Verzeichniseinträgen ist eine der von Benutzeranwendungen am häufigsten ausgeführten Operationen. Es kann auch die hinsichtlich des Leistungsumfangs am besten sichtbare

Operation sein. Folglich wird viel Aufmerksamkeit darauf gerichtet, dafür zu sorgen, daß das Verzeichnis wirksam durchsucht wird, und das Dateisystem von WindowsNT (NTFS) dupliziert ausreichend Datei-Inoden-Informationen in dem Verzeichniseintrag, so daß eine Verzeichnisleseoperation erfüllt werden kann, indem die Verzeichniseinträge durchsucht und gelesen werden, ohne daß dieses verlassen wird, um die Informationen von den Datei-Inoden zu lesen. Das mit diesem Schema verbundene Problem besteht darin, daß die doppelt gespeicherten Datei-Metadaten, wie die Dateizeitmarkierungen und die Dateigröße, häufig schnell aktualisiert werden können, wodurch das Aktualisieren der Metadaten kostspieliger gemacht wird. Dieser Zusatzaufwand wird jedoch in Hinblick auf die Leistungsfähigkeit als akzeptierbar angesehen, die bei Verzeichnisdurchsuchungsoperationen gewonnen wird.

[0055] Das Dateisystem **60** verwendet die gleiche Philosophie des Bereitstellens einer wirksamen Verzeichnisdurchsuchung, indem Datei-Inoden-Informationen in Verzeichniseinträgen dupliziert werden. Jeder Verzeichniseintrag enthält ausreichend Informationen, um die Dateiinformations-Abfrageanforderungen von Win32 zu erfüllen. Der Datei-Inode ist mit den Dateistrom-Deskriptoren auf einer getrennten Seite gespeichert. Der Inode wird über einen Zeiger im Verzeichniseintrag lokalisiert.

[0056] Die Verzeichniseinträge des Dateisystems werden in dem Verzeichniseintrags-Datenstrom der Verzeichnisdatei gespeichert. Zum Maximieren der Platzausnutzung wird jeder Verzeichniseintrag dem ersten verfügbaren freien Platz auf einer Seite zugeordnet, der den gesamten Eintrag aufnehmen kann. Die Länge des Eintrags hängt von der Länge des primären Namens der Datei ab. Die folgenden Informationen sind Teil des Verzeichniseintrags: die Erzeugungszeit, die Änderungszeit, die letzte Schreibzeit, die letzte Zugriffszeit, Zeiger auf den Strom-Deskriptor, der Zeiger auf den Stammverzeichnis-Inoden, MS-DOS-Dateiattribute, der MS-DOS-Dateiname (8.3-Namenskonvention). Für durchschnittliche Dateinamenslängen enthält eine Seite bis zu etwa 30 Einträge. Alle Dateiinformationen in dem Verzeichniseintrag sind, mit Ausnahme des primären Namens der Datei und des MS-DOS-Dateinamens, auch in dem Datei-Inoden enthalten. Die primären Dateinamen und zugeordnete kurze Namen sind nur in den Verzeichniseinträgen gespeichert. Hierdurch wird die Größe des Inodens feststehend gemacht.

[0057] Wenn eine Dateiinformation modifiziert wird (mit Ausnahme von Dateinamen), wird der Inode in Zusammenhang mit dem Aktualisierungsvorgang aktualisiert und enthält daher stets die aktuellsten Informationen. Die zugeordnete Verzeichniseintragsänderung wird locker geräumt, um die Kosten einer Doppelaktualisierung zu verringern. Dies bedeutet, daß die Inoden-Aktualisierungen entweder geräumt werden oder wiederherstellbar sind, jedoch nicht die entsprechenden Verzeichniseintragsaktualisierungen. Falls der Verzeichniseintrag die Synchronität mit dem Inoden verliert (wenn die Inoden-Änderung erfolgreich geräumt wird, jedoch nicht die Verzeichnisänderung), wird der Eintrag bei der nächsten Aktualisierung des Inodens aktualisiert. Um die Synchronisation von Verzeichnisaktualisierungen zu erleichtern, können die Verzeichniseinträge (Inoden) nicht mehrere Seiten umfassen. **Fig.** 3 zeigt die Organisation von Verzeichniseinträgen und zugeordneten Inoden.

DATEIEN

[0058] Eine Datei des Dateisystems **60** weist Datenströme und die Dateisystem-Metadaten zum Beschreiben der Datei auf. Dateien werden in dem Dateisystem **60** durch als Inoden bezeichnete Objekte beschrieben. Der Inode ist eine Datenstruktur, die Datei-Metadaten speichert. Er repräsentiert die Datei in dem Dateisystem **60**.

[0059] Ein Datenstrom ist ein logisch zusammenhängender Bytestrom. Er kann aus den von Anwendungen gespeicherten Daten oder den von dem Dateisystem **60** gespeicherten internen Informationen bestehen. Die Datenströme werden auf Seiten abgebildet, die zum Speichern von dem adressierbaren gemeinsam verwendeten Speicherplatz **20** zugeordnet werden. Das Dateisystem **60** segmentiert einen Datenstrom in eine Folge von 4-Kilobyte-Segmenten, wobei jedes Segment einer Seite entspricht. Das Dateisystem **60** enthält zwei Größeninformationsteile je Datenstrom, nämlich die Anzahl von Bytes im Datenstrom und die Zuordnungsgröße in Seitenanzahlen. Der Bytestrom zum Segmentieren bzw. seitenweisen Anordnen von Abbildungsinformationen ist Teil der Datei-Metadaten und wird in einer als Datenstrom-Deskriptor bezeichneten Struktur gespeichert. Siehe **Fig.** 4.

[0060] Benutzeranforderungen von Daten werden in bezug auf den Bytebereich und die Position des Anfangsbytes, gemessen durch seinen Versatz gegenüber dem Anfang des Datenstroms, der Byteposition null, spezifiziert. Das Dateisystem **60** bildet den Versatz in die Seite, die das Anfangsbyte enthält, und den Versatz innerhalb der Seite vom Anfang der Seite an ab.

[0061] Jede Datei des Dateisystems **60** weist wenigstens zwei Datenströme auf, nämlich den Standard-Datenstrom und den Zugriffssteuerlisten-Strom (Access Control List Stream – ACL Stream). Jede Datei kann wahlweise auch andere Datenströme aufweisen. Der ACL-Strom wird verwendet, um den Sicherheits-Zugriffssteuerlisten-Satz zu der Datei zu speichern. Jeder Datenstrom wird einzeln bezeichnet, so daß der Benutzer einen spezifischen Datenstrom erzeugen oder den Zugriff auf diesen öffnen kann. Es wird angenommen, daß der Name des Standard-Datenstroms der primäre Name der Datei ist. Zum Zugreifen auf einen Datenstrom muß der Benutzer des Dateisystems **60** zuerst eine Datei-Handle

auf den gewünschten Datenstrom dem Namen nach öffnen. Falls der Dateiname verwendet wird, wird die Handle für den Standard-Datenstrom geöffnet. Diese Dateiöffnungs-Handle stellt den Datenstrom in allen Diensten des Dateisystems dar, welche auf den Datenstrom einwirkt.

[0062] Das Dateisystem **60** exportiert einen Satz von Diensten, um auf dem Dateiniveau zu arbeiten. Die Eingabe in die Dienste sind die Dateiobjekt-Handle (Inode) oder die Datenstrom-Objekt-Handle und die Operations-spezifischen Parameter, einschließlich der gewünschten Abschnitte des Datenstroms in Bytepositionen.

[0063] Offene Dateien werden durch Datenstromobjekte (oder einfach Dateiobjekte) dargestellt. Benutzer greifen auf Dateien unter Verwendung dieser Dateiobjekte zu, die von den Benutzern durch Datei-Handles identifiziert werden. Eine Datei-Handle ist eine 32-Bit-Einheit, die eine Ausprägung eines offenen Dateistroms darstellt. Beispielsweise erzeugt Windows NT das Dateiobjekt und gibt ansprechend auf die Benutzeranforderung einer Dateierzeugung oder einer Dateiöffnung eine Datei-Handle an die Benutzer zurück. Das Dateisystem **60** initialisiert einen Zeiger auf einen Dateisteuerblock. Mehrere Dateiobjekte zeigen auf denselben Dateisteuerblock, und jeder Dateisteuerblock enthält getrennte Stromobjekte für jeden Öffnungskontext. Extern ist die Datei-Handle für die Benutzer undurchsichtig. Mehrere Öffnungsanweisungen können an dieselbe Datei ausgegeben werden. Wenn der Benutzer eine Datei schließt, werden das Dateiobjekt und die zugeordnete Datei-Handle entfernt.

[0064] Das Dateisystem **60** bildet Dateiströme in Segmentfolgen ab, die zunehmend größer werden, wobei jedes Segment einer oder mehreren Seiten entspricht. Das Dateisystem **60** versucht, aufeinanderfolgende Seiten für Datenströme zu reservieren, es weist realen Externspeicher jedoch nur dann zu, wenn dies notwendig ist, gewöhnlich als Ergebnis einer Dateierweiterung, die erforderlich ist, wenn über die Zuordnungsgröße des Datenstroms hinaus geschrieben wird. Wenn eine Dateierweiterungsanforderung empfangen wird, rundet das Dateisystem **60** die Erweiterungsgröße in der Anzahl der Bytes auf ein Vielfaches von 4 Kilobytes auf, um sie zu einer ganzen Seitenzahl zu machen, und fordert Seiten zur tatsächlichen Zuordnung an. Die Anzahl der vom Dateisystem zugeordneten 4-Kilobyte-Seiten hängt von der Anzahl der vorgenommenen Dateierweiterungsanforderungen ab. Das Dateisystem **60** ordnet eine 4-Kilobyte-Seite für die erste Erweiterungsanforderung, zwei 4-Kilobyte-Seiten für die zweite Anforderung, vier 4-Kilobyte-Seiten für die dritte Erweiterungsanforderung usw. zu. Die neu zugeordneten Seiten werden mit Nullen aufgefüllt. Durch Reservieren aufeinanderfolgender Seiten kann das Dateisystem **60** den Umfang der Buchhaltungsinformationen auf den Byte-Versatz für die Seitenabbildung reduzieren. Das Dateisystem **60** reserviert mehr als den erforderlichen Speicherplatz für eine Datei (manchmal viel mehr) und substanziiert den Speicher durch Seite für Seite erfolgendes Zuordnen von Externspeicher.

[0065] Es werden Vier-Kilobyte-Zuordnungssegmente gewählt, um den nicht verwendeten Speicherplatz zu verringern und dennoch eine vernünftige Zuordnungsgröße für gewöhnliche Dateierweiterungen bereitzustellen. Weil eine Zuordnung eine kostspielige Operation ist (die sehr wahrscheinlich verteilte Operationen beinhaltet), ist eine kleinere Zuordnungsgröße nicht wirksam. Eine höhere Zuordnungsgröße würde zu einer unwirksamen Platzausnutzung oder zu einer zusätzlichen Komplexität beim Verwalten ungenutzten Platzes führen. Ein 4-Kilobyte-Segment wird auch natürlich auf eine Seite abgebildet, wodurch das Datenstromsegment für die Seitenabbildung vereinfacht wird. Wenngleich eine Analogie mit der Zuordnungspolitik von 4-Kilobyte-Clustergrößen (Segmentgrößen) von NTFS für große Platten zum Beschleunigen des Zuordnens und zum Verringern der Fragmentierung gebildet werden könnte, ist eine solche Analogie nicht vollständig gültig, weil die tatsächliche Zuordnungssegmentgröße auf der Platte in hohem Maße von der lokalen Plattengröße und den physikalischen Dateisystemen abhängt.

[0066] Ähnlich dem NTFS, das die Zuordnung jeder Plattenpartition steuert und daher schnell den für die Zuordnung verfügbaren freien Platz auf dem Datenträger bestimmen kann, fordert das Dateisystem **60** Informationen zu dem gesamten verfügbaren Platz an und verwendet diese Informationen zum schnellen Bestimmen, ob die Zuordnungsverarbeitung fortgesetzt werden soll. Falls der verfügbare Gesamtplatz kleiner ist als die erforderliche Zuordnungsgröße, wird die Anforderung sofort abgelehnt. Andernfalls geht das Dateisystem **60** dazu über, die Seiten zuzuordnen, um die Anforderung zu erfüllen. Die Tatsache, daß das Dateisystem **60** mit der Zuordnung fortfahren kann, garantiert nicht, daß die Zuordnung gelingt, weil sich der tatsächlich insgesamt verfügbare Platz ständig ändern kann.

[0067] Das Dateisystem **60** nutzt die Seitenebenen-Replikationsfähigkeit des zugrundeliegenden verteilt adressierbaren gemeinsam verwendeten Speichersystems **20** aus, die in der US-Patentanmeldung offenbart ist, auf die vorstehend verwiesen wurde. Die Replikation auf der Seitenebene ermöglicht, daß das System eine Dateireplikation bereitstellt. Die Datenströme einer replizierten Datei werden durch Seiten unterstützt, die selbst repliziert werden. Auf diese Weise werden Datenströme ohne Eingriff des Dateisystems **60** automatisch repliziert. Der zusätzliche Platz, der von den mehreren Replika verbraucht wird, spiegelt sich in den Dateigrößen (Datenstromgrößen) nicht wieder. Die Stromzuordnungsgröße teilt weiterhin die gesamte Zuordnungsgröße in Seiten mit, die für eine Replik erforderlich ist. Die temporäre Dateien unterstützenden Seiten werden jedoch nicht repliziert.

DATEIZUGRIFF UND BETRIEBSMITTEL-GEMEINSAMVERWENDUNG – SPERRE

[0068] Der gemeinsam verwendete Speicher bildet den Verteilungsmechanismus zur Betriebsmittel-Gemeinsamverwendung zwischen Peer-Knoten, auf denen die Software des Dateisystems **60** läuft. Jede Ausprägung des Dateisystems **60** auf jedem Netzwerkknoten betrachtet die gemeinsam verwendeten Speicherbetriebsmittel (also Seiten) als mit anderen lokalen oder fernen Teilprozessen gemeinsam verwendet. Das Dateisystem **60** braucht einen Weg zum Implementieren von Dateisystem-Sperren hoher Ebene, um eine konsistente Betriebsmittel-Gemeinsamverwendung bereitzustellen. Jede beliebige Zeitgleichszugriffs-Steuerstruktur kann verwendet werden, um Sperren, wie Sperrobjekte oder Semaphoren, zu implementieren. Bei Datenbankanwendungen kann eine Sperre auch durch Implementieren von Zeitgleichszugriffs-Steuerstrukturen erreicht werden, die Datenbankindizes oder – schlüsseln zugeordnet sind. Bei Dateisystemanwendungen kann der Zugriff auf Dateien oder Verzeichnisse gesteuert werden. Ein weiteres Beispiel von Dateisystem-Sperren ist das Bytebereichssperren; das den Benutzern die Möglichkeit gibt, einen gemeinsamen Zugriff auf Dateien zu koordinieren. Eine Bytebereichssperre ist ein Sperrensatz für einen Bytebereich einer Datei. Ein koordinierter gemeinsamer Zugriff auf eine Datei kann erreicht werden, indem die gewünschten Bytebereiche mit Sperren versehen werden. Im allgemeinen funktioniert die Dateisystem-Sperre hoher Ebene in der folgenden Weise: (a) Ein Dateisystem-Betriebsmittel ist von jeder Ausprägung des Dateisystems 60 gemeinsam zu verwenden, und der Zugriff auf das Betriebsmittel wird durch ein Sperrprotokoll unter Verwendung einer Sperrobjekt-Datenstruktur koordiniert, die die Sperre hoher Ebene darstellt, um das gemeinsam verwendete Betriebsmittel zu koordinieren, und der Wert der Datenstruktur stellt den aktuellen Zustand der Sperre dar, (b) zum Zugreifen auf das Betriebsmittel muß die Ausprägung an jedem Knoten in der Lage sein, den Zustand (oder den Wert) der Sperrdatenstruktur zu betrachten, und falls sie "frei" ist, sie modifizieren, so daß sie "belegt" wird, falls sie jedoch "belegt" ist, muß sie warten, bis sie "frei" wird, und es könnten Zwischenzustände zwischen "frei" und "belegt" (also mehr als zwei Sperrzustände) vorhanden sein, in jedem Fall ist bei diesem Bytebereichs-Sperrbeispiel eine Sperre jedoch eine Beschreibung eines bestimmten Bytebereichs, der von irgendeinem Teilprozeß des Dateisystems **60** geteilt verwendet bzw. ausschließlich gesperrt wird, und eine einen Konflikt erzeugende neue Bytebereichs-Sperranforderung, die in den bereits gesperrten Bytebereich fällt oder diesen überlagert, wird abgelehnt, oder der Requester kann blockieren (abhängig davon, wer die Anforderung gemacht hat), und (c) der Zugriff auf die Sperrdatenstruktur oder eine Modifikation davon durch die Ausprägung jedes Knotens muß seriell angeordnet werden, so daß er dann wiederum verwendet werden kann, um die Betriebsmittel-Gemeinsamverwendung hoher Ebene zu koordinieren.

[0069] Die Sperrmerkmale und -fähigkeiten der in der US-Patentanmeldung mit der laufenden Nummer 08/754 481 beschriebenen gemeinsam verwendeten Speichermaschine ermöglichen es dem Dateisystem **60**, den Zugriff auf Seiten zu koordinieren. Die Maschine kann auch verwendet werden, um den Zugriff auf Betriebsmittel zu koordinieren, im Fall einer komplexen Betriebsmittelsperre hoher Ebene in der Art einer Bytebereichssperre kann die direkte Verwendung der Sperrmerkmale und -fähigkeiten zum Bereitstellen von Sperren jedoch aus den folgenden Gründen zu kostspielig sein: (a) jede Bytebereichssperre würde eine Seite benötigen, die die Sperre darstellt, und weil die Anzahl der Bytebereichssperren groß sein kann, können die Kosten in Hinblick auf den Seitenverbrauch zu hoch sein, und (b) die Maschinensperren bieten nur zwei Sperrzustände (nämlich gemeinsam verwendet und exklusiv), und Dateisystem-Sperren hoher Ebene können mehr Sperrzustände benötigen.

[0070] Das Dateisystem **60** der Erfindung implementiert die Dateisystem-Sperre unter Verwendung der Maschinensperre als ein Grundelement bzw. ein primitives Element zum Bereitstellen einer Serialisierung zum Zugreifen auf die Sperrdatenstrukturen und zum Aktualisieren von diesen. Zum Lesen einer Sperrstruktur nimmt das Dateisystem **60** eine gemeinsame Sperre auf der Seite der Datenstruktur unter Verwendung der Merkmale und Fähigkeiten der Maschinensperre, bevor es die Seite liest, um zu verhindern, daß die Datenstruktur modifiziert wird. Zum Modifizieren der Sperrstruktur setzt es eine exklusive Sperre auf die Seite. Die Seitensperre wird fortgenommen und gelöst, sobald der Sperrstrukturwert gelesen oder modifiziert wird.

[0071] Mit der von der Seitensperr- und Seitenungültigkeitsmachungs-Benachrichtigung bereitgestellten Serialisierung implementiert das Dateisystem **60** die Sperren hoher Ebene in der folgenden Weise: (a) Zum Nehmen einer Dateisystem-Sperre (FS-Sperre) setzt das Dateisystem **60** eine geteilte Sperre auf die FS-Sperrseite und liest die Seite und untersucht dann die Sperrstruktur, (b) falls die Sperrstruktur angibt, daß das Betriebsmittel entsperrt ist oder in einem kompatiblen Sperrmodus gesperrt ist, fordert das Dateisystem **60** das exklusive Sperren der Seite an, wodurch garantiert wird, daß nur eine Knotenausprägung des Dateisystems **60** die Sperrdatenstruktur modifizieren kann, und falls die Anforderung Erfolg hat, führt das Dateisystem **60** eine Schreibabbildung der Sperrseite aus und ändert dann die Sperrstruktur, um die Sperre zu setzen und entsperrt die Seite und setzt den Seitenzugriff auf nicht vorhanden, und (c) falls das Betriebsmittel in einem inkompatiblen Sperrmodus gesperrt wird, entsperrt das Dateisystem **60** die Seite, hält die gelesene Seite jedoch im abgebil-

deten Zustand und gibt sich (den aktuellen Teilprozeß) dann in eine Warteschleife und wartet auf ein Systemereignis, das mitteilt, daß sich der Sperrwert geändert hat, und wenn sich der Sperrwert tatsächlich ändert, wird der Teilprozeß des Dateisystems **60** benachrichtigt und wiederholt den vorstehenden Schritt (a). Das Dateisystem **60** implementiert die Mitteilung unter Verwendung eines primitiven Signalelements. Die auf eine Sperre wartenden Teilprozesse des Dateisystems **60** werden bei einem Systemereignis blockiert.

[0072] Wenn sich die Seite, die die Sperre enthält, ändert, wird ein Signal zu jedem blockierten Teilprozeß des Dateisystems **60** gesendet. Alle blockierten Teilprozesse des Dateisystems **60** wachen dann auf und wiederholen Schritt (a). Dateisystem-Sperren werden in flüchtigen Seiten gespeichert.

DATEIZUGRIFF UND BETRIEBSMITTEL-GEMEINSAMVERWENDUNG – BYTEBEREICHS-SPERRE

[0073] Eine Bytebereichs-Sperre ist ein Dateisystem-Sperrdienst, der über die LockFile()- und die LockFileEx()-API von Win32 zu den Benutzern exportiert wird. Sie ermöglicht einen gleichzeitigen Zugriff auf verschiedene nicht überlappende Bereiche eines Dateidatenstroms durch mehrere Benutzer. Zum Zugreifen auf den Datenstrom sperrt der Benutzer den Bereich (Bytebereich) der Datei, um einen exklusiven oder gemeinsamen Lesezugriff auf den Bereich zu erhalten.

[0074] Das Dateisystem **60** unterstützt Bytebereichs-Sperren für jeden einzelnen Datenstrom der Datei. Das folgende Bytebereichs-Sperrverhalten nach Art von Win32 wird unterstützt: (a) Das Sperren eines Bereichs einer Datei wird verwendet, um einen gemeinsamen oder exklusiven Zugriff auf den spezifizierten Bereich der Datei zu erhalten, und das Dateisystem **60** verfolgt Bytebereichssperren durch die Datei-Handle, weshalb Datei-Handles eine Möglichkeit bereitstellen, den Eigentümer der Sperre eindeutig zu identifizieren, (b) das Sperren eines Bereichs, der über die aktuelle Dateiendeposition hinausgeht, ist kein Fehler, (c) das Sperren eines Teils einer Datei für einen exklusiven Zugriff verweigert allen anderen Prozessen sowohl einen Lese- als auch einen Schreibzugriff auf den spezifizierten Bereich der Datei, und das Sperren eines Teils einer Datei für einen gemeinsamen Zugriff verweigert allen anderen Prozessen den Schreibzugriff auf den spezifizierten Bereich der Datei, ermöglicht es jedoch anderen Prozessen, den gesperrten Bereich zu lesen, was bedeutet, daß das Dateisystem **60** Bytebereichssperren, die für den Datenstrom festgelegt sind, nicht nur auf Sperranforderungen, sondern auf jeden Lese- oder Schreibzugriff prüfen muß, (d) falls eine exklusive Sperre für einen Bereich gefordert wird, der bereits entweder gemeinsam oder exklusiv von anderen Teilprozessen gesperrt ist, blockiert die Anforderung sofort oder schlägt sofort fehl, wobei dies von der spezifizierten Rufoption abhängt, und (e) Sperren dürfen einen existierenden gesperrten Bereich der Datei nicht überlappen.

[0075] Für jede Bytebereichssperre erzeugt das Dateisystem **60** einen Bytebereichs-Sperrdatensatz zum Darstellen der Sperre. Der Datensatz enthält die folgenden Informationen (a) einen Bytebereich, (b) einen Sperrmodus (gemeinsam verwendet oder exklusiv), (c) eine Prozeßidentifikation und (d) einen Win32-Sperrschlüsselwert.

[0076] Das Dateisystem **60** sieht die Datei-Bytebereiche als Betriebsmittel mit einem gesteuerten Zugriff an. Für jeden Bytebereichs-Sperrdatensatz erzeugt das Dateisystem **60** eine Dateisystem-Sperre (wie vorstehend erörtert wurde), um den Zugriff auf das Bytebereichs-"Betriebsmittel" zu koordinieren. Eine kompatible Bytebereichs-Sperranforderung (gemeinsame Sperre) führt dazu, daß Lesesperren von der Dateisystem-Sperre genommen werden, die dem Bytebereichs-Datensatz zugeordnet ist. Eine exklusive Bytebereichs-Sperranforderung wird abgebildet, um eine Schreibsperre von der Dateisystem-Sperre zu nehmen.

[0077] Unter Verwendung des vorstehend erörterten Dateisystem-Sperrmechanismus werden Sperranforderungen mitgeteilt, die auf der Seite warten, die den gewünschten Bytebereich enthält, wenn sich der Seiteninhalt ändert.

Adressierbarer gemeinsam verwendeter Speicherplatz

[0078] Nachdem die Erfindung und verschiedene Ausführungsformen davon in einigen Einzelheiten beschrieben wurden, wird nun eine detailliertere Beschreibung des adressierbaren gemeinsam verwendeten Speicherplatzes gegeben, der in der am 22. November 1996 eingereichten anliegenden US-Patentanmeldung mit der laufenden Nummer 08/754 481 beschrieben ist. Alle nachstehend bereitgestellten Informationen sind in dieser Patentanmeldung enthalten.

[0079] Das in der US-Patentanmeldung, auf die hiermit verwiesen sei, beschriebene adressierbare gemeinsam verwendete Speichersystem ist eine "Maschine", die einen virtuellen Speicherplatz erzeugen und verwalten kann, der von allen Computern auf einem Netzwerk gemeinsam verwendet werden kann und den Speicherplatz aller mit dem Netzwerk verbundener Speichervorrichtungen umfassen kann. Demgemäß können alle auf dem Netzwerk gespeicherten Daten innerhalb des virtuellen Speicherplatzes gespeichert werden, und der tatsächliche physikalische Ort der Daten kann in beliebigen der mit dem Netzwerk verbundenen Speichervorrichtungen liegen.

[0080] Insbesondere kann die Maschine oder das System ein globales Adressensignal, das einen beispielsweise 4 Kilobytes umfassenden Teil des virtuellen Speicherplatzes darstellt, erzeugen oder empfan-

gen. Das globale Adressensignal kann von den physikalischen Plätzen und Bezeichnerplätzen der zugrundeliegenden Computerhardware entkoppelt sein, also nicht in Beziehung dazu stehen, um einen Speicherplatz zu unterstützen, der groß genug ist, um jede flüchtige und permanente Speichervorrichtung, die mit dem System verbunden ist, einzuschließen. Beispielsweise können Systeme der Erfindung auf 32-Bit-Computern arbeiten, sie können jedoch auch globale Adressensignale verwenden, die 128 Bytes breit sind. Demgemäß umspannt der virtuelle Speicherplatz 2^{128} Bytes, was viel größer ist als der 2^{32} Bytes umfassende Adressenplatz, der von der zugrundeliegenden Computerhardware unterstützt wird. Ein solcher Adressenplatz kann groß genug sein, um eine getrennte Adresse für jedes Byte an Datenspeicher auf dem Netzwerk, einschließlich aller RAM-, Platten- und Bandspeicher, bereitzustellen.

[0081] Für einen solchen großen virtuellen Speicherplatz speichert zu jeder Zeit typischerweise nur ein kleiner Teil Daten. Dementsprechend weist das System eine Verzeichnisverwaltungseinheit auf, die jene Abschnitte des virtuellen Speicherplatzes verfolgt, die verwendet werden. Das System stellt für jeden Abschnitt des verwendeten virtuellen Speicherplatzes physikalischen Speicher bereit, indem es alle diese Abschnitte auf eine physikalische Speichervorrichtung in der Art eines RAM-Speichers oder einer Festplatte abbildet. Wahlweise weist die Abbildung eine Indirektheitsebene auf, wodurch eine Datenmigration, ein fehlertoleranter Betrieb und ein Lastausgleich ermöglicht werden.

[0082] Indem ermöglicht wird, daß jeder Computer überwacht und verfolgt, welche Abschnitte des virtuellen Speicherplatzes verwendet werden, kann jeder Computer den Speicherplatz geteilt verwenden. Dies ermöglicht es, daß es so aussieht, daß vernetzte Computer einen einzigen Speicher aufweisen, und es kann daher dadurch ermöglicht werden, daß Anwendungsprogramme, die auf verschiedenen Computern laufen, unter Verwendung von Techniken kommunizieren, die gegenwärtig zum Kommunizieren zwischen auf derselben Maschine laufenden Anwendungen verwendet werden.

[0083] Gemäß einem Aspekt kann verstanden werden, daß die Erfindung aus der vorstehend identifizierten US-Patentanmeldung, auf die verwiesen wurde, Computersysteme einschließt, die einen adressierbaren gemeinsam verwendeten Speicherplatz aufweisen. Die Systeme können ein Datennetzwerk, das computerlesbare Informationen darstellende Datensignale überträgt, eine permanente Speichervorrichtung, die mit dem Datennetzwerk gekoppelt ist und die einen permanenten Datenspeicher bereitstellt und mehrere Computer, die jeweils eine Schnittstelle aufweisen, die mit dem Datennetzwerk gekoppelt ist, um auf das Datennetzwerk zuzugreifen und damit Datensignale auszutauschen, aufweisen. Weiterhin kann jeder der Computer ein gemeinsam verwendetes Speicheruntersystem zum Abbilden eines Abschnitts des adressierbaren Speicherplatzes auf einen Abschnitt des permanenten Speichers aufweisen, um eine adressierbare permanente Speicherung von Datensignalen bereitzustellen.

[0084] Es wird verständlich sein, daß bei einem System, das den Speicher über die Speichervorrichtungen des Netzwerks verteilt, die permanente Speichervorrichtung eine Mehrzahl an lokalen permanenten Speichervorrichtungen aufweist, die jeweils mit einem jeweiligen der mehreren Computer gekoppelt sind. Hierzu kann das System auch einen Verteiler zum Abbilden von Abschnitten des adressierbaren Speicherplatzes über die Mehrzahl lokaler permanenter Speichervorrichtungen und eine Plattenverzeichnis-Verwaltungseinheit zum Verfolgen der abgebildeten Abschnitte des adressierbaren Speicherplatzes aufweisen, um Informationen bereitzustellen, die die lokale permanente Speichervorrichtung repräsentieren, in der dieser Abschnitt des darauf abgebildeten adressierbaren Speicherplatzes gespeichert wird.

[0085] Das System kann auch ein Cache-System aufweisen, um eine der lokalen permanenten Speichervorrichtungen als einen Cache-Speicher zu betreiben. der dazu dient, eine Cache-Speicherung von Datensignalen vorzunehmen, die Abschnitten des adressierbaren Speicherplatzes zugeordnet sind, auf die vor kurzem zugegriffen wurde. Weiterhin kann das System eine Migrationssteuereinrichtung zum selektiven Bewegen von Abschnitten des adressierbaren Speicherplatzes zwischen den lokalen permanenten Speichervorrichtungen der mehreren Computer aufweisen. Die Migrationssteuereinrichtung kann Datenzugriffsmuster, Betriebsmittelanforderungen oder andere Kriterien oder Heuristiken zur Verwirklichung mit der Erfindung bestimmen und auf diese reagieren. Demgemäß kann die Migrationssteuereinrichtung die Lasten auf dem Netzwerk ausgleichen und Daten zu Knoten bewegen, an denen gemeinsam auf sie zugegriffen wird. Die Cache-Steuereinrichtung kann ein Softwareprogramm sein, das auf einem Hostcomputer läuft, um einen durch Software verwalteten RAM und einen Platten-Cache bereitzustellen. Der RAM kann ein beliebiger flüchtiger Speicher unter Einschluß von SRAM, DRAM oder eines anderen flüchtigen Speichers sein. Die Platte kann ein beliebiger permanenter Speicher unter Einschluß einer beliebigen Platten-, RAID-, Band- oder anderen Vorrichtung sein, der einen permanenten Datenspeicher bereitstellt.

[0086] Das System kann auch eine kohärente Replikationssteuereinrichtung zum Erzeugen einer Kopie oder einer ausgewählten Anzahl von Kopien eines Abschnitts des in der lokalen permanenten Speichervorrichtung eines ersten Computers enthaltenen adressierbaren Speicherplatzes und zum Speichern der Kopie in der lokalen permanenten Speichervorrichtung eines zweiten Computers aufweisen. Die kohärente Replikationssteuereinrichtung kann die Kohärenz der Kopien erhalten, um eine kohärente Da-

tenreplikation bereitzustellen.

[0087] Es wird auch verständlich, daß die Systeme eine integrierte Steuerung in einem flüchtigen Speicher und in einem permanenten Speicher gespeicherter Daten bereitstellen. Bei solchen Systemen weist eine flüchtige Speichervorrichtung einen flüchtigen Speicher für Datensignale auf, und das gemeinsam verwendete Speicheruntersystem weist ein Element, typischerweise ein Softwaremodul, zum Abbilden eines Abschnitts des adressierbaren Speicherplatzes auf einen Abschnitt des flüchtigen Speichers auf. Bei diesen Systemen kann die flüchtige Speichervorrichtung aus mehreren lokalen flüchtigen Speichervorrichtungen bestehen, die jeweils mit einem jeweiligen der mehreren Computer gekoppelt sind, und die permanente Speichervorrichtung kann aus mehreren lokalen permanenten Speichervorrichtungen bestehen, die jeweils mit einem jeweiligen der mehreren Computer gekoppelt sind.

[0088] Bei diesen Systemen kann eine Verzeichnisverwaltungseinheit die abgebildeten Abschnitte des adressierbaren Speicherplatzes verfolgen und zwei Unterkomponenten, nämlich eine Plattenverzeichnis-Verwaltungseinheit zum Verfolgen von Abschnitten des adressierbaren Speicherplatzes, die auf die lokalen permanenten Speichervorrichtungen abgebildet sind, und eine RAM-Verzeichnisverwaltungseinheit zum Verfolgen von Abschnitten des adressierbaren Speicherplatzes, die auf die lokalen flüchtigen Speichervorrichtungen abgebildet sind, aufweisen. Wahlweise kann ein RAM-Cache-System eine der lokalen flüchtigen Speichervorrichtungen als einen Cache-Speicher zum Cache-Speichern von Datensignalen, die Abschnitten des adressierbaren Speicherplatzes zugeordnet sind, auf die vor kurzem zugegriffen wurde, betreiben.

[0089] Die Systeme können zusätzliche Elemente einschließlich eines Seitenwechselelements zum erneuten Abbilden eines Abschnitts des adressierbaren Speicherplatzes zwischen einer der lokalen flüchtigen Speichervorrichtungen und einer der lokalen permanenten Speichervorrichtungen, einer Verfahrenssteuereinrichtung zum Bestimmen eines Betriebsmittel-Verfügbarkeitssignals, das auf jedem der mehreren Computer verfügbaren Speicher darstellt, eines Seitenwechselelements, das den Abschnitt des adressierbaren Speicherplatzes von einer Speichervorrichtung eines ersten Computers ansprechend auf das Betriebsmittel-Verfügbarkeitssignal auf eine Speichervorrichtung eines zweiten Computers erneut abbildet, und einer Migrationssteuereinrichtung zum Bewegen von Abschnitten des adressierbaren Speicherplatzes zwischen den lokalen flüchtigen Speichervorrichtungen der mehreren Computer aufweisen.

[0090] Wahlweise können die Systeme eine Hierarchieverwaltungseinrichtung zum Organisieren der mehreren Computer zu einem Satz hierarchischer Gruppen aufweisen, wobei jede Gruppe wenigstens einen der mehreren Computer aufweist. Jede der Gruppen kann eine Gruppen-Speicherverwaltungseinrichtung aufweisen, um Abschnitte des adressierbaren Speicherplatzes als Funktion der hierarchischen Gruppen zu migrieren.

[0091] Das System kann die Kohärenz zwischen kopierten Abschnitten des Speicherplatzes durch Aufnehmen einer kohärenten Replikationssteuereinrichtung zum Erzeugen einer kohärenten Kopie eines Abschnitts des adressierbaren Speicherplatzes aufrechterhalten.

[0092] Das System kann globale Adressensignale erzeugen oder empfangen. Dementsprechend können die Systeme einen Adressengenerator zum Erzeugen eines globalen Adressensignals aufweisen, das einen Abschnitt des adressierbaren Speicherplatzes darstellt. Der Adressengenerator kann eine Überbrückungseinheit aufweisen, um globale Adressensignale als Funktion einer den permanenten Speichervorrichtungen zugeordneten Speicherkapazität zu erzeugen, um globale Adressensignale bereitzustellen, die geeignet sind, die Speicherkapazität der permanenten Speichervorrichtungen logisch zu adressieren.

[0093] Bei verteilten Systemen kann die Verzeichnisverwaltungseinheit eine verteilte Verzeichnisverwaltungseinheit zum Speichern eines eine Speicherstelle eines Abschnitts des adressierbaren Speicherplatzes darstellenden Verzeichnissignals innerhalb des verteilten Speicherplatzes sein. Die verteilte Verzeichnisverwaltungseinheit kann einen Verzeichnisseitengenerator zum Zuordnen eines Abschnitts des adressierbaren Speicherplatzes und zum Speichern eines Eintragssignals, das einen Abschnitt des Verzeichnissignals darstellt, aufweisen. Der Verzeichnisseitengenerator weist wahlweise einen Bereichsgenerator zum Erzeugen eines einen Abschnitt des adressierbaren Speicherplatzes darstellenden Bereichssignals und zum Erzeugen des auf das Bereichssignal ansprechenden Eintragssignals auf, um ein Eintragssignal bereitzustellen, das einen Abschnitt des Verzeichnissignals darstellt, der dem Abschnitt des adressierbaren Speicherplatzes entspricht. Weiterhin kann die verteilte Verzeichnisverwaltungseinheit ein Verknüpfungssystem zum Verknüpfen der Verzeichnisseiten, um eine hierarchische Datenstruktur der verknüpften Verzeichnisseiten zu bilden, sowie ein Bereichsverknüpfungssystem zum Verknüpfen der Verzeichnisseiten als Funktion des Bereichssignals, um eine hierarchische Datenstruktur der verknüpften Verzeichnisseiten zu bilden, aufweisen.

[0094] Weil die von dem System gespeicherten Daten in der Hinsicht heimatlos sein können, daß die Daten keine feste physikalische Heimat haben, sondern je nach dem, wie es die Betriebsmittel und andere Faktoren vorschreiben, zwischen den Speichervorrichtungen des Netzwerks migrieren können, kann ein Computersystem gemäß der Erfindung einen Verzeichnisseitengenerator einschließen, der einen Knotenwähler aufweist, um ein verantwortliches

Knotensignal zu erzeugen, das einen ausgewählten der mehreren Computer darstellt, die Ortsinformationen für einen Abschnitt des gemeinsam verwendeten Adressenplatzes aufweisen. Hierdurch wird eine Indirektheitsebene bereitgestellt, die das Verzeichnis von dem physikalischen Speicherplatz der Daten entkoppelt. Dementsprechend braucht das Verzeichnis nur den Knoten oder eine andere Vorrichtung zu identifizieren, die den physikalischen Ort der Daten verfolgt. Auf diese Weise braucht das Verzeichnis nicht jedesmal aktualisiert zu werden, wenn Daten zwischen physikalischen Speicherplätzen migrieren, weil der den Ort der Daten verfolgende Knoten nicht gewechselt hat und noch die physikalischen Ortsinformationen bereitstellt.

[0095] Demgemäß kann das System Seitengeneratoren aufweisen, die Verzeichnisseiten erzeugen, die eine Ortsüberwachungseinrichtung darstellende Informationen in der Art eines verantwortlichen Computerknotens, der eine Datenspeicherstelle verfolgt, aufweisen, um eine Verzeichnisstruktur zum Verfolgen heimatloser Daten bereitzustellen. Weiterhin kann das Verzeichnis selbst in Form von Seiten innerhalb des virtuellen Speicherplatzes gespeichert werden. Daher kann die Datenspeicherstelle eine Verzeichnisseite darstellende Informationen speichern, um die Verzeichnisstruktur als Seiten heimatloser Daten zu speichern.

[0096] Gemäß einem weiteren Aspekt kann die Erfindung der vorstehend identifizierten US-Patentanmeldung, auf die hiermit verwiesen sei, als Verfahren zum Versehen eines Computersystems mit einem adressierbaren gemeinsam verwendeten Speicherplatz verstanden werden. Das Verfahren kann die Schritte des Bereitstellens eines Netzwerks zum Übertragen computerlesbare Informationen darstellender Datensignale, des Bereitstellens einer Festplatte, die mit dem Netzwerk gekoppelt ist und einen permanenten Speicher für Datensignale aufweist, des Bereitstellens mehrerer Computer, die jeweils eine mit dem Datennetzwerk gekoppelte Schnittstelle zum Austauschen von Datensignalen zwischen den mehreren Computern aufweisen, und des Zuweisens eines Abschnitts des adressierbaren Speicherplatzes zu einem Abschnitt des permanenten Speichers der Festplatte zum Bereitstellen eines adressierbaren permanenten Speichers für Datensignale aufweisen.

[0097] Mit Bezug auf die Zeichnungsbestandteile, die sich auf das adressierbare gemeinsam verwendete Speichersystem oder die Maschine der vorstehend identifizierten US-Patentanmeldung beziehen, auf die hier verwiesen wird, sei bemerkt, daß **Fig.** 5 ein Computernetzwerk **10** darstellt, das einen gemeinsam verwendeten Speicher bereitstellt, der den Speicherplatz von jedem der Knoten des dargestellten Computernetzwerks **210** umfaßt.

[0098] Insbesondere zeigt **Fig.** 5 ein Computernetzwerk **210**, das eine Mehrzahl von Knoten **212a**–**212c** umfaßt, die jeweils eine CPU **214**, ein Betriebssystem **216**, eine optionale private Speichervorrichtung **218** und ein gemeinsam verwendetes Speicheruntersystem **220** aufweist. Wie weiter in **Fig.** 5 dargestellt ist, ist jeder Knoten **212a**–**212c** über das gemeinsam verwendete Speicheruntersystem **220** mit einem virtuellen gemeinsam verwendeten Speicher **222** verbunden. Wie nachstehend in näheren Einzelheiten erklärt wird, ermöglicht das Computernetzwerk **210** durch Bereitstellen des gemeinsam verwendeten Speicheruntersystems **220**, das den Knoten **212a**–**212c** ermöglicht, auf den virtuellen gemeinsam verwendeten Speicher **222** zuzugreifen, daß die Netzwerkknoten **212a**–**212c** unter Verwendung derselben Techniken, die von Anwendungen verwendet werden, wenn sie zwischen Anwendungen kommunizieren, die auf derselben Maschine laufen, kommunizieren und Funktionalitäten gemeinsam verwenden. Diese Techniken können eine Objektverknüpfung und -einbettung, dynamische Verknüpfungsbibliotheken, eine Klassenregistrierung und andere solche Techniken verwenden. Demgemäß können die Knoten **212** den virtuellen gemeinsam verwendeten Speicher **222** verwenden, um Daten und Objekte zwischen Anwendungsprogrammen auszutauschen, die auf den verschiedenen Knoten **212** des Netzwerks **210** laufen.

[0099] Bei der in **Fig.** 5 dargestellten Ausführungsform kann jeder Knoten **212** ein herkömmliches Computersystem, wie bspw. ein im Handel erhältliches IBM-PC-kompatibles Computersystem, sein. Der Prozessor **214** kann eine beliebige Prozessoreinheit sein, die geeignet ist, um die Datenverarbeitung für dieses Computersystem auszuführen. Das Betriebssystem **216** kann ein beliebiges im Handel erhältliches oder firmeneigenes Betriebssystem sein, das Funktionen zum Zugreifen auf den lokalen Speicher des Computersystems und Netzwerkfunktionen aufweist oder auf diese zugreifen kann.

[0100] Die private Speichervorrichtung **218** kann eine beliebige Computerspeichervorrichtung sein, die dafür geeignet ist, computerlesbare Informationen darstellende Datensignale zu speichern. Der private Speicher versorgt den Knoten mit lokalem Speicher, der für die anderen Knoten auf dem Netzwerk unzugänglich gehalten werden kann. Typischerweise weist die private Speichervorrichtung **218** einen RAM oder einen Abschnitt eines RAM-Speichers auf, um Daten und Anwendungsprogramme vorübergehend zu speichern und um dem Prozessor **214** mit Speicher zum Ausführen von Programmen zu versorgen. Die private Speichervorrichtung **18** kann auch einen permanenten Speicher, typischerweise eine Festplatteneinheit oder einen Abschnitt einer Festplatteneinheit aufweisen, um Daten permanent zu speichern.

[0101] Das gemeinsam verwendete Speicheruntersystem **220**, das in **Fig.** 5 dargestellt ist, ist eine Ausführungsform der Erfindung, die eine Kopplung zwischen dem Betriebssystem **216** und dem virtuellen gemeinsam verwendeten Speicher **222** herstellt und eine Schnittstelle zwischen dem Betriebssystem **216** und dem virtuellen gemeinsam verwendeten Spei-

cher bildet, um es dem Betriebssystem **216** zu ermöglichen, auf den virtuellen gemeinsam verwendeten Speicher **222** zuzugreifen. Das dargestellte gemeinsam verwendete Speicheruntersystem **220** ist ein Softwaremodul, das als eine alleinstehende, verteilte gemeinsam verwendete Speichermaschine arbeitet. Das dargestellte System dient der Erläuterung, und es können auch andere Systeme gemäß der Erfindung als gemeinsam verwendete Speicheruntersysteme verwirklicht werden, die in ein Anwendungsprogramm eingebettet werden können oder als ein eingebetteter Code einer Hardwarevorrichtung implementiert werden können. Andere solche Anwendungen können verwirklicht werden, ohne von dem Schutzumfang der Erfindung abzuweichen.

[0102] Der dargestellte virtuelle gemeinsam verwendete Speicher **222** zeigt einen virtuellen gemeinsam verwendeten Speicher, der von jedem der Knoten **212a**–**212c** über das gemeinsam verwendete Speicheruntersystem **220** zugänglich ist. Der virtuelle gemeinsam verwendete Speicher **222** kann auf Vorrichtungen abbilden, die einen physikalischen Speicher für vom Computer lesbare Daten bereitstellen, die in **Fig.** 5 als eine Mehrzahl von Seiten **224a**–**224d** dargestellt sind. Bei einer Ausführungsform bilden die Seiten Abschnitte des gemeinsam verwendeten Speicherplatzes, und sie unterteilen den Adressenplatz des gemeinsam verwendeten Speichers in seitenadressierbare Speicherplätze.

[0103] Beispielsweise kann der Adressenplatz in 4-Kilobyte-Abschnitte seitenweise abgebildet werden. Bei anderen Ausführungsformen kann eine alternative Granularität verwendet werden, um den gemeinsam verwendeten Speicherplatz zu verwalten. Jeder Knoten **212a**–**212c** des gemeinsam verwendeten Speicheruntersystems **220** kann auf jede im virtuellen gemeinsam verwendeten Speicher **222** gespeicherte Seite **224a**–**224d** zugreifen. Jede Seite **224a**–**224d** stellt einen eindeutigen Eintrag innerhalb des virtuellen gemeinsam verwendeten Speichers **222** gespeicherter Computerdaten dar. Jede Seite **224a**–**224d** ist für jeden der Knoten **212a**–**212c** zugänglich, und jeder Knoten kann alternativ zusätzliche Datenseiten innerhalb des virtuellen gemeinsam verwendeten Speichers **222** speichern. Jede neu gespeicherte Datenseite kann für jeden der anderen Knoten **212a**–**212c** zugänglich sein. Demgemäß liefert der virtuelle gemeinsam verwendete Speicher **222** ein System zum Gemeinsamverwenden und Übertragen von Daten zwischen jedem Knoten **212** des Computernetzwerks **210**.

[0104] **Fig.** 6 zeigt in Form eines Funktionsblockdiagramms ein Computernetzwerk **230**, das einen verteilten gemeinsam verwendeten Speicher aufweist. Bei dieser Ausführungsform weist jeder Knoten **212a**–**212c** ein Speicheruntersystem **232** auf, das zwischen dem Betriebssystem **216** und den zwei lokalen Speichervorrichtungen, dem RAM **234** und der Platte **236** verbindet und das weiterhin mit einem Netzwerk **238** koppelt, das mit jedem der dargestellten Knoten **212a**, **212b** und **212c** und einer Netzwerk-Speichervorrichtung **226** koppelt.

[0105] Insbesondere zeigt **Fig.** 6 ein verteiltes gemeinsam verwendetes Speichernetzwerk **30**, das eine Mehrzahl von Knoten **212a**–**212c** aufweist, die jeweils eine Verarbeitungseinheit **214**, ein Betriebssystem **216**, ein Speicheruntersystem **232**, einen RAM **234** und eine Platte **236** aufweisen. **Fig.** 6 zeigt weiterhin ein Computernetzwerksystem **38**, das zwischen den Knoten **212a**–**212c** und der Netzwerk-Speichervorrichtung **226** verbindet. Das Netzwerk **238** liefert ein Netzwerkkommunikationssystem über diese Elemente.

[0106] Die dargestellten Speicheruntersysteme **232a**–**232c**, die zwischen dem Betriebssystem **216a**–**216c**, den Speicherelementen **234a**–**234c**, **236a**–**236c** und dem Netzwerk **238** verbinden, kapseln die lokalen Speicher aller Knoten ein, um eine Abstraktion eines gemeinsam verwendeten virtuellen Speichersystems bereitzustellen, das alle Knoten **212a**– **212c** auf dem Netzwerk **238** umfaßt. Die Speicheruntersysteme **232a**–**232c** können Softwaremodule sein, die als Verteiler wirken, um Abschnitte des adressierbaren Speicherplatzes über die dargestellten Speichervorrichtungen abzubilden. Die Speicheruntersysteme verfolgen weiterhin die im lokalen Speicher jedes Knotens **212** gespeicherten Daten und betreiben weiterhin Netzwerkverbindungen mit dem Netzwerk **238**, um Daten zwischen den Knoten **212a**–**212c** zu übertragen. Auf diese Weise greifen die Speicheruntersysteme **232a**–**232c** auf jedes Speicherelement auf dem Netzwerk **238** zu und steuern dieses, um Speicherzugriffsoperationen auszuführen, die für das Betriebssystem **216** transparent sind. Demgemäß verknüpft das Betriebssystem **216** als eine Schnittstelle zu einem globalen Speicherplatz, der alle Knoten **212a**–**212c** auf dem Netzwerk **238** umfaßt, mit dem Speicheruntersystem **232**.

[0107] **Fig.** 6 zeigt weiterhin, daß das System **230** einen verteilten gemeinsam verwendeten Speicher bereitstellt, der einen permanenten Speicher für Abschnitte des verteilten Speichers einschließt. Insbesondere weist die dargestellte Ausführungsform ein Speicheruntersystem, wie bspw. ein Untersystems **232a**, auf, das mit einer als Platte **236a** dargestellten permanenten Speichervorrichtung verknüpft. Das Untersystem **232a** kann die permanente Speichervorrichtung betreiben, um einen permanenten Speicher für Abschnitte des verteilten gemeinsam verwendeten Speicherplatzes bereitzustellen. Wie dargestellt ist, ist auf jede der in **Fig.** 6 dargestellten permanenten Speichervorrichtungen **236** ein Abschnitt des adressierbaren Speicherplatzes abgebildet. Beispielsweise sind auf die Vorrichtung **236a** die Abschnitte des C_o, C_d, C_g des adressierbaren Speicherplatzes abgebildet, und sie bietet einen permanenten Speicher für in diesen Adressenbereichen gespeicherte Datensignale.

[0108] Dementsprechend kann das Untersystem **232a** eine integrierte Steuerung permanenter Spei-

chervorrichtungen und eines elektronischen Speichers bereitstellen, um zu ermöglichen, daß der verteilte gemeinsam verwendete Speicherplatz beide Typen von Speichervorrichtungen umfaßt und um zu ermöglichen, daß Abschnitte des verteilten gemeinsam verwendeten Speichers, abhängig von vorbestimmten Bedingungen, wie der jüngsten Verwendung, zwischen dem permanenten und dem elektronischen Speicher bewegt werden.

[0109] Bei einer optionalen Ausführungsform sind die Knoten des Netzwerks zu einer Hierarchie von Gruppen organisiert. Bei dieser Ausführungsform können die Speicheruntersysteme **232a** –**232c** eine Hierarchieverwaltungseinrichtung aufweisen, die eine hierarchische Steuerung für die Verteilung von Daten bereitstellt. Dies umfaßt das Steuern der Migrationssteuereinrichtung und der Verfahrenssteuereinrichtung, die nachstehend in näheren Einzelheiten erörtert werden, zum Ausführen einer hierarchischen Datenmigration und eines Lastausgleichs, so daß Daten in erster Linie zwischen Computern derselben Gruppe migrieren und in hierarchischer Ordnung an andere Gruppen übergeben werden. Die Betriebsmittelverteilung wird in ähnlicher Weise verwaltet.

[0110] **Fig.** 7 zeigt in näheren Einzelheiten ein gemeinsam verwendetes Speicheruntersystem **240** gemäß der Erfindung. **Fig.** 7 zeigt ein gemeinsam verwendetes Speicheruntersystem **240**, das eine Schnittstelle **242**, eine DSM-Verzeichnisverwaltungseinheit **244**, eine Speichersteuereinrichtung **246**, eine lokale Platten-Cache-Steuereinrichtung **248** und eine lokale RAM-Cache-Steuereinrichtung **250** aufweist. **Fig.** 7 zeigt weiterhin das Netzwerk **254**, einen optionalen Kunden des DSM-Systems, der als der Dienst **258** dargestellt ist, das Betriebssystem **216**, einen Plattentreiber **260**, ein Plattenelement **262** und ein RAM-Element **264**.

[0111] Das in **Fig.** 7 dargestellte gemeinsam verwendete Speicheruntersystem **240** kann die Speicherverwaltungsoperationen des Netzwerkknotens **212** einschließen, um einen virtuellen gemeinsam verwendeten Speicher bereitzustellen, der jeden Knoten umfassen kann, der sich mit dem Netzwerk **254** verbindet. Dementsprechend betrachtet jeder lokale Knoten **212** das Netzwerk als einen Knotensatz, die jeweils mit einem großen gemeinsam verwendeten Computerspeicher verbunden sind.

[0112] Die dargestellte Schnittstelle **242** bildet einen Eintrittspunkt für den lokalen Knoten, um auf den gemeinsam verwendeten Speicherplatz des Computernetzwerks zuzugreifen. Die Schnittstelle **242** kann direkt mit dem Betriebssystem **216**, einer verteilten Diensteinrichtung, wie bspw. das dargestellte DSM-Dateisystem **258**, einer verteilten Diensteinrichtung, auf der Benutzerebene oder alternativ einer Kombination davon gekoppelt werden.

[0113] Die dargestellte Schnittstelle **242** stellt eine API bereit, die eine speicherorientierte API ist. Demgemäß kann die dargestellte Schnittstelle **242** einen Satz von Schnittstellen exportieren, die eine Steuerung des verteilten Speichers auf einer niedrigen Ebene bereitstellen. Wie in **Fig.** 7 dargestellt ist, exportiert die Schnittstelle **242** die API zu dem Betriebssystem **216** oder zu dem optionalen DSM-Dienst **258**. Das Betriebssystem **216** oder der Dienst verwendet die Schnittstelle **242**, um Standard-Speicherverwaltungstechniken, wie bspw. ein Lesen oder Schreiben von Abschnitten des Speicherplatzes, anzufordern. Diese Abschnitte des Speicherplatzes können die vorstehend beschriebenen Seiten sein, die 4-Kilobyte-Abschnitte des gemeinsam verwendeten Speicherplatzes oder andere Speichereinheiten, wie Objekte oder Segmente, sein können. Jede Seite kann sich innerhalb des gemeinsam verwendeten Speicherplatzes befinden, und sie wird durch ein globales Adressensignal für diese Seite des Speichers bezeichnet. Das System kann Adressensignale von einem Anwendungsprogramm empfangen oder optional einen globalen Adressengenerator aufweisen, der die Adressensignale erzeugt. Der Adressengenerator kann ein Bereichsmodul aufweisen, das Adressensignale für einen Speicherplatz erzeugt, der die Speicherkapazität des Netzwerks umfaßt.

[0114] Demgemäß empfängt die Schnittstelle **242** bei einer Ausführungsform Anforderungen zum Manipulieren von Seiten des gemeinsam verwendeten Speicherplatzes. Hierzu kann die Schnittstelle **242** ein Softwaremodul aufweisen, das eine Bibliothek von Funktionen aufweist, die von Diensten, dem OS **216** oder einem anderen Rufer oder einer anderen Vorrichtung aufgerufen werden können. Die Funktionsaufrufe stellen dem OS **216** eine API speicherorientierter Dienste hoher Ebene, wie Daten lesen, Daten schreiben und Speicher zuordnen, bereit. Die Implementation der Funktionen kann einen Satz von Aufrufen von Steuerungen einschließen, welche die Verzeichnisverwaltungseinheit **244** und die lokale Speichersteuereinrichtung **246** betreiben. Dementsprechend kann die Schnittstelle **242** ein Satz von Speicherfunktionsaufrufen hoher Ebene zur Verknüpfung mit den Funktionselementen niedriger Ebene des gemeinsam verwendeten Speicheruntersystems **240** sein.

[0115] **Fig.** 7 zeigt weiterhin eine DSM-Verzeichnisverwaltungseinheit **244**, die mit der Schnittstelle **242** gekoppelt ist. Die Schnittstelle **242** übergibt Anforderungssignale, die Anforderungen zum Implementieren von Speicheroperationen, wie das Zuordnen eines Speicherabschnitts, das Sperren eines Speicherabschnitts, das Abbilden eines Speicherabschnitts oder eine andere solche Speicherfunktion, darstellen. Die Verzeichnisverwaltungseinheit **244** verwaltet ein Verzeichnis, das Abbildungen aufweisen kann, die jede mit dem in **Fig.** 6 dargestellten Netzwerk **238** verbundene Speichervorrichtung, einschließlich jedes RAM- und Plattenelements, das für das Netzwerk zugänglich ist, umfassen können. Die Verzeichnisverwaltungseinheit **244** speichert eine globale Verzeichnisstruktur, die eine Abbildung des globalen Adressenplatzes bereitstellt. Bei einer Ausführungs-

form, die nachstehend in näheren Einzelheiten erklärt wird, stellt die Verzeichnisverwaltungseinheit **244** ein globales Verzeichnis bereit, das zwischen globalen Adressensignalen und verantwortlichen Knoten auf dem Netzwerk abbildet. Ein verantwortlicher Knoten speichert Informationen hinsichtlich des Orts und von Attributen von Daten, die einer jeweiligen globalen Adresse zugeordnet sind, und er speichert optional eine Kopie der Daten dieser Seite. Folglich verfolgt die Verzeichnisverwaltungseinheit **244** Informationen zum Zugreifen auf jede beliebige Adressenstelle innerhalb des Bezeichnerraums.

[0116] Die Steuerung des verteilten gemeinsam verwendeten Speichers kann durch die Verzeichnisverwaltungseinheit **244** und die Speichersteuereinrichtung **246** koordiniert werden. Die Verzeichnisverwaltungseinheit **244** unterhält eine Verzeichnisstruktur, die mit einer von der Schnittstelle **242** empfangenen globalen Adresse zusammenwirken kann und für diese Adresse einen Knoten auf dem Netzwerk identifizieren kann, der dafür verantwortlich ist, die dieser Adresse des gemeinsam verwendeten Speicherplatzes zugeordnete Seite zu unterhalten. Sobald die Verzeichnisverwaltungseinheit **244** identifiziert, welcher Knoten für das Unterhalten einer bestimmten Adresse verantwortlich ist, kann die Verzeichnisverwaltungseinheit **244** einen Knoten identifizieren, der Informationen zum Lokalisieren einer Kopie der Seite speichert, und den Aufruf an die Speichersteuereinrichtung **246** dieses Knotens richten und an die Speichersteuereinrichtung dieses Knotens die von der Speicherschnittstelle **242** bereitgestellte Speicheranforderung übergeben. Dementsprechend ist die dargestellte Verzeichnisverwaltungseinheit **244** dafür verantwortlich, eine Verzeichnisstruktur zu verwalten, die für jede Seite des gemeinsam verwendeten Speicherplatzes einen verantwortlichen Knoten identifiziert, der den physikalischen Ort der auf der jeweiligen Seite gespeicherten Daten verfolgt. Demgemäß kann das Verzeichnis, statt direkt den Ort der Seite bereitzustellen, optional einen verantwortlichen Knoten oder eine andere Vorrichtung identifizieren, die den Ort der Seite verfolgt. Diese Indirektheit erleichtert das Unterhalten des Verzeichnisses, wenn Seiten zwischen Knoten migrieren.

[0117] Die Speichersteuereinrichtung **246** führt die Speicherzugriffsfunktionen niedriger Ebene aus, die physikalisch Daten innerhalb der mit dem Netzwerk verbundenen Speicherelemente speichern. Bei der dargestellten Ausführungsform kann die Verzeichnisverwaltungseinheit **244** eines ersten Knotens eine Speicherzugriffsanforderung über die Schnittstelle **242** an das Netzwerkmodul des OS **216** und über das Netzwerk **254** zu einem zweiten Knoten leiten, den die Verzeichnisverwaltungseinheit **244** als den verantwortlichen Knoten für die gegebene Adresse identifiziert. Die Verzeichnisverwaltungseinheit **244** kann dann den verantwortlichen Knoten abfragen, um die Attribute und den gegenwärtigen Eigentümerknoten der Speicherseite, die der jeweiligen globalen Adresse zugeordnet ist, zu bestimmen. Der Eigentümer der jeweiligen Seite ist der Netzwerkknoten, der eine Kontrolle über das Speicherelement hat, auf dem die Daten der zugeordneten Seite gespeichert sind. Die Speichersteuereinrichtung **246** des Eigentümers kann über das OS **216** dieses Knotens oder über irgendeine Schnittstelle auf den Speicher des Eigentümerknotens zugreifen, um auf die Daten der Seite zuzugreifen, die physikalisch auf diesem Eigentümerknoten gespeichert sind.

[0118] Insbesondere ist die Verzeichnisverwaltungseinheit **244**, wie in **Fig.** 7 dargestellt ist, mit dem Netzwerkmodul **252** gekoppelt, das mit dem Netzwerk **254** gekoppelt ist. Die Verzeichnisverwaltungseinheit kann zum Netzwerkmodul **252** einen Befehl und zugeordnete Daten übertragen, welche die Netzwerkschnittstelle **252** anweisen, ein Datensignal an den Eigentümerknoten zu übergeben. Der Eigentümerknoten empfängt die Speicheranforderung über das Netzwerk **254** und über das Netzwerkmodul **252**, das die Speicheranforderung an die Schnittstelle **242** dieses Eigentümerknotens übergibt. Die Schnittstelle **242** ist mit der Speichersteuereinrichtung **246** gekoppelt und kann die Speicheranforderung an die lokale Speichersteuereinrichtung dieses Eigentümerknotens übergeben, um die lokalen Speicherelemente, wie die Platten- oder RAM-Elemente, zu betreiben und die angeforderte Speicheroperation auszuführen.

[0119] Sobald der Eigentümerknoten die angeforderte Speicheroperation, wie das Lesen einer Datenseite, ausgeführt hat, kann das Speicheruntersystem **240** des Eigentümerknotens die Datenseite übertragen oder eine Kopie der Datenseite über das Netzwerk **254** zu dem Knoten übertragen, der ursprünglich den Zugriff auf diesen Abschnitt des gemeinsam verwendeten Speichers angefordert hat. Die Datenseite wird über das Netzwerk **254** zu dem Netzwerkmodul **252** des anfordernden Knotens übertragen, und das gemeinsam verwendete Speicheruntersystem **240** veranlaßt die Speichersteuereinrichtung **246**, in dem lokalen Speicher des anfordernden Knotens eine Kopie der Daten, auf die zugegriffen wurde, zu speichern.

[0120] Demgemäß identifiziert die Verzeichnisverwaltungseinheit **244** bei einer Ausführungsform der Erfindung, wenn ein erster Knoten auf eine Seite des gemeinsam verwendeten Speicherplatzes zugreift, die nicht lokal an diesem Knoten gespeichert ist, einen Knoten, der eine Kopie der auf dieser Seite gespeicherten Daten aufweist, und verschiebt eine Kopie dieser Daten in den lokalen Speicher des anfordernden Knotens. Der lokale Speicher, sowohl der flüchtige als auch der permanente, des anfordernden Knotens wird daher zu einem Cache-Speicher für Seiten, die von diesem lokalen Knoten angefordert worden sind. Diese Ausführungsform ist in **Fig.** 7 dargestellt, die eine Speichersteuereinrichtung zeigt, die eine lokale Platten-Cache-Steuereinrichtung **248** und eine lokale RAM-Cache-Steuereinrichtung **250** auf-

weist. Diese beiden lokalen Cache-Steuereinrichtungen können für das Betriebssystem **216** oder einen anderen Kunden Seiten des gemeinsam verwendeten Speicherplatzes bereitstellen, die indem lokalen Speicher des Knotens, einschließlich des lokalen permanenten Speichers und des lokalen flüchtigen Speichers, Cache-artig gespeichert sind.

[0121] Das gemeinsam verwendete Speicheruntersystem kann eine kohärente Replikationssteuereinrichtung aufweisen, die die Kohärenz zwischen in dem Cache-Speicher abgelegten Seiten aufrechterhält, indem sie eine Kohärenz durch einen Ungültigmachungsprozeß, eine Kohärenz durch einen Migrationsprozeß oder einen anderen Kohärenzprozeß verwendet, der zur Verwirklichung mit der vorliegenden Erfindung geeignet ist. Die kohärente Replikationssteuereinrichtung kann automatisch eine Kopie der auf jeder Seite gespeicherten Daten erzeugen und die Kopie in einer Speichervorrichtung speichern, die von der Speichervorrichtung der ursprünglichen Kopie getrennt ist. Hierdurch wird ein fehlertoleranter Betrieb bereitgestellt, weil der Fehler irgendeiner Speichervorrichtung nicht zu einem Datenverlust führt. Die kohärente Replikationssteuereinrichtung kann ein Softwaremodell sein, das alle im flüchtigen Speicher gehaltenen und für das Schreiben verfügbar gemachten Seitenkopien überwacht. Die Steuereinrichtung kann beliebige der vorstehend erwähnten Kohärenztechniken verwenden und Tabellen von Ortsinformationen speichern, die die Ortsinformationen für alle erzeugten Kopien identifizieren.

[0122] **Fig.** 8 zeigt in näheren Einzelheiten eine Ausführungsform eines gemeinsam verwendeten Speicheruntersystems gemäß der Erfindung. Das in **Fig.** 8 dargestellte gemeinsam verwendete Speicheruntersystem **270** weist ein Fernoperationselement **274**, einen lokalen RAM-Cache-Speicher **276**, einen RAM-Kopiersatz **278**, ein globales RAM-Verzeichnis **280**, einen Platten-Kopiersatz **282**, ein globales Plattenverzeichnis **284**, eine Konfigurationsverzeichniseinheit **288**, ein Verfahrenselement **290** und einen lokalen Platten-Cache-Speicher **94** auf. Weiterhin sind in **Fig.** 8 ein Netzwerkelement **304**, ein physikalischer Speicher **300**, ein gemeinsam verwendetes Datenelement **302**, ein physikalisches Dateisystem **298**, das Teil des Betriebssystems **216** ist, ein Konfigurationsdienst **308**, ein diagnostischer Dienst **310** und eine Speicherzugriffsanforderung **312** dargestellt. Das dargestellte Untersystem **270** kann ein Computerprogramm sein, das mit dem physikalischen Speicher, dem Dateisystem und dem Netzwerksystem des Hostknotens gekoppelt ist, oder es kann aus elektrischen Schaltungskartenanordnungen bestehen, die mit dem Hostknoten verknüpft sind, oder es kann eine Kombination von Programmen und Schaltungskartenanordnungen sein.

[0123] Die in **Fig.** 8 dargestellte Ablaufplanungseinrichtung **272** kann die von einer API des Untersystems **270** bereitgestellten Steuerungen koordinieren. Gemäß einer Ausführungsform kann die Ablaufplanungseinrichtung **272** eine Zustandsmaschine sein, die die Anforderungen **312** und die Fernanforderungen über das Netzwerk **304** überwacht und auf diese reagiert, welche Anweisungen für Speicheroperationen sein können und welche die globalen Adressen, die verarbeitet werden, darstellende Signale einschließen können. Diese Speicheroperationsanforderungen **312** können als Operationscodes für primitive Operationen an einer oder mehreren globalen Adressen wirken. Sie können Anforderungen oder andere Speicheroperationen lesen und schreiben. Alternativ kann die Ablaufplanungseinrichtung **272** ein Programm, wie bspw. ein Interpreter sein, das eine Ausführungsumgebung bereitstellt und diese Operationscodes in als Applets bezeichnete Steuerflußprogramme abbilden kann. Die Applets können unabhängig ausführbare Programme sein, die sowohl Umgebungsdienste, wie eine Unterteilung in Teilprozesse, eine Synchronisation und eine Pufferverwaltung, als auch die in **Fig.** 8 dargestellten Elemente verwenden. Die API kann sowohl von externen Clients wie ein verteiltes gemeinsam verwendetes Speicherdateisystem als auch rekursiv von den Applets und den anderen Elementen **274** –**294** des Untersystems **270** aufgerufen werden. Jedes Element kann eine Verkapselungsebene für die Verwaltung eines bestimmten Betriebsmittels oder eines Aspekts des Systems bereitstellen. Hierzu kann jedes Element eine API exportieren, die aus von den Applets zu verwendenden Funktionen besteht. Diese Struktur ist in **Fig.** 8 dargestellt. Dementsprechend kann die Ablaufplanungseinrichtung **272** eine Umgebung zum Laden und Ausführen von Applets bereitstellen. Die Applets werden von der Ablaufplanungseinrichtung **272** auf einer Je-Operationscode-Basis abgefertigt und können die Ablaufsteuerung für die sequentielle oder parallele Ausführung eines Elements ausführen, um den Operationscode an der spezifizierten globalen Adresse in der Art einer Lese- oder Schreiboperation zu implementieren. Wahlweise kann die Ablaufplanungseinrichtung **272** ein Element zum dynamischen Ändern des Ablaufs während der Laufzeit aufweisen und Applets in einem parallelen und einem interpretierten Modus ausführen.

[0124] Das dargestellte gemeinsam verwendete Speicheruntersystem **270** weist eine verzweigte Verzeichnisverwaltungseinheit auf, die ein globales RAM-Verzeichnis **280** und das globale Plattenverzeichnis **284** aufweist. Das globale RAM-Verzeichnis **280** ist eine Verzeichnisverwaltungseinheit, die Informationen verfolgen kann, die den Ort der Seiten liefern können, die im flüchtigen Speicher, typischerweise dem RAM, oder den Netzwerkknoten gespeichert sind. Das globale Plattenverzeichnis **284** ist eine globale Platten-Verzeichnisverwaltungseinheit, die eine Verzeichnisstruktur verwaltet, die Informationen verfolgt, die den Ort von Seiten bereitstellen können, die auf permanenten Speichervorrichtungen gespeichert werden. Zusammen bilden das globale RAM-Ver-

zeichnis **280** und das globale Plattenverzeichnis **284** das gemeinsam verwendete Speicheruntersystem **270** mit einer integrierten Verzeichnisverwaltung für Seiten, die in dem permanenten Speicher und in dem flüchtigen Speicher gespeichert sind.

[0125] Bei einer Ausführungsform kann ein Seitenwechselelement die RAM- und die Platten-Verzeichnisverwaltungseinheit einsetzen, um Abschnitte des adressierbaren Speicherplatzes zwischen einem der flüchtigen Speicher und einem der permanenten Speicher neu abzubilden. Bei dem gemeinsam verwendeten Speichersystem ermöglicht dies, daß das Seitenwechselelement Seiten aus dem flüchtigen Speicher eines Knotens auf einen Plattenspeicher eines anderen Knotens neu abbildet. Demgemäß übergibt die RAM-Verzeichnisverwaltungseinheit die Steuerung dieser Seite an die Platten-Verzeichnisverwaltungseinheit, die die Seite dann wie jede andere Datenseite behandeln kann. Dies ermöglicht einen verbesserten Lastausgleich durch Entfernen von Daten aus dem RAM-Speicher und ein Speichern von ihnen in den Plattenvorrichtungen unter der Steuerung durch die Platten-Verzeichnisverwaltungseinheit.

[0126] Die lokale Speichersteuereinrichtung des Untersystems **270** ist durch den lokalen RAM-Cache-Speicher **276** und den lokalen Platten-Cache-Speicher **294** gebildet. Der lokale RAM-Cache-Speicher **276**, der mit dem physikalischen Speicher **300** des lokalen Knotens gekoppelt ist, kann, wie vorstehend beschrieben wurde, auf den virtuellen Speicherplatz des lokalen Knotens zugreifen, um auf Daten zuzugreifen, die physikalisch innerhalb des RAM-Speichers **300** gespeichert sind. In ähnlicher Weise ist der lokale Platten-Cache-Speicher **294** mit der permanenten Speichervorrichtung **298** gekoppelt und kann auf eine physikalische Stelle zugreifen, die in dem lokalen permanenten Speicher Daten des verteilten gemeinsam verwendeten Speichers enthält.

[0127] **Fig.** 8 zeigt auch ein Fernoperationselement **274**, das das Netzwerk **304** und die Ablaufplanungseinrichtung **272** koppelt. Das Fernoperationselement **274** verhandelt die Übertragung von Daten über das Netzwerk **304**, um Abschnitte der im gemeinsam verwendeten Speicherplatz gespeicherten Daten zwischen den Knoten des Netzwerks zu übertragen. Das Fernoperationselement **274** kann auch Dienste von fernen Peers anfordern, also eine Ungültigmachung vornehmen, um dabei zu helfen, die Kohärenz aufrechtzuerhalten, oder dies aus anderen Gründen tun.

[0128] **Fig.** 8 zeigt auch ein Verfahrenselement **290**, das ein Softwaremodul sein kann, das als eine Steuereinrichtung zum Feststellen der Verfügbarkeit von Betriebsmitteln, wie Druckerfähigkeiten, Festplattenplatz, verfügbarem RAM und anderen solchen Betriebsmitteln, wirkt. Die Verfahrenssteuereinrichtung kann beliebige der geeigneten Heuristiken einsetzen, um die Elemente in der Art der Seitenwechsel-Steuereinrichtung, der Platten-Verzeichnisverwaltungseinheit und anderer Elemente anzuweisen, die verfügbaren Betriebsmittel dynamisch zu verteilen.

[0129] **Fig.** 8 zeigt weiterhin ein Speicheruntersystem **270**, das einen RAM-Kopiersatz **278** und einen Platten-Kopiersatz **282** aufweist. Diese Kopiersätze können Kopien von Seiten verwalten, die an einem einzigen Knoten in dem Cache-Speicher abgelegt sind. Der Platten-Kopiersatz **282** kann Informationen zu Seitenkopien enthalten, die in dem lokalen Platten-Cache-Speicher gespeichert sind, der der lokale permanente Speicher sein kann. In ähnlicher Weise kann der RAM-Kopiersatz **278** Informationen zu Seitenkopien enthalten, die in dem lokalen RAM-Cache-Speicher, der der lokale RAM sein kann, abgelegt sind. Diese Kopiersätze schließen das Indexieren und das Speichern von Kopiersatzdaten ein, die von Applets oder anderem Ausführungscode verwendet werden können, um die Kohärenz in dem gemeinsam verwendeten Speicherplatz gespeicherter Daten aufrechtzuerhalten. Die Kopiersatzelemente können Kopiersatzdaten enthalten, die die vom Hostknoten Cache-artig gespeicherten Seiten identifizieren. Weiterhin kann der Kopiersatz die anderen Knoten auf dem Netzwerk identifizieren, die eine Kopie dieser Seite enthalten; und er kann weiterhin für jede Seite identifizieren, welcher dieser Knoten der Eigentümerknoten ist, wobei der Eigentümerknoten ein Knoten sein kann, der Schreibprivilegien für die Seite, auf die zugegriffen wird, aufweist. Die Kopiersätze selbst können auf Seiten des verteilten gemeinsam verwendeten Speicherplatzes gespeichert werden.

[0130] Der lokale RAM-Cache-Speicher **276** liefert Speicher für Speicherseiten und ihre Attribute. Bei einer Ausführungsform liefert der lokale RAM-Cache-Speicher **276** einen globalen Adressenindex zum Zugreifen auf die Cache-artig gespeicherten Seiten des verteilten Speichers und die auf dieser Seite beruhenden Attribute. Bei dieser Ausführungsform liefert der lokale RAM-Cache-Speicher **276** den Index durch Speichern einer Liste jeder in dem lokalen RAM Cache-artig abgelegten globalen Adresse durch Eingeben in den Speicher. Mit jeder aufgelisteten globalen Adresse liefert der Index einen Zeiger in einen Pufferspeicher und auf den Ort der Seitendaten. Wahlweise kann der Index mit jeder aufgelisteten globalen Adresse weiterhin Attributinformationen bereitstellen, die ein Versionskennzeichen, das die Version der Daten darstellt, eine Speichermarke, die darstellt, ob die im RAM abgelegten Daten eine Kopie der auf der Platte gespeicherten Daten sind oder ob die in dem RAM abgelegten Daten modifiziert worden sind, jedoch noch nicht auf die Platte übertragen worden sind, ein Flüchtigkeitsbit zum Angeben, ob die Seite durch einen Reservespeicher in dem permanenten Speicher hinterlegt ist, und andere solche Attributinformationen, die nützlich sind, um die Kohärenz der gespeicherten Daten zu verwalten, einschließen.

[0131] Bei der in **Fig.** 8 dargestellten Ausführungsform liefert das Speicheruntersystem **270** den Knotenzugriff auf den verteilten Speicherplatz durch die

koordinierte Operation der Verzeichnisverwaltungseinheit, die das globale RAM-Verzeichnis **280** und das globale Plattenverzeichnis **284** aufweist, der Cache-Steuereinrichtung, die den lokalen RAM-Cache-Speicher und die lokalen Platten-Cache-Elemente **276** und **294** aufweist, und der Kopiersatzelemente, die den RAM-Kopiersatz **278** und den Platten-Kopiersatz **282** aufweisen.

[0132] Die Verzeichnisverwaltungseinheit liefert eine Verzeichnisstruktur, die den gemeinsam verwendeten Adressenplatz indexiert. Um das Beispiel eines seitenweise organisierten gemeinsam verwendeten Adressenplatzes fortzusetzen sei bemerkt, daß die Verzeichnisverwaltungseinheit des Untersystems **270** dem Hostknoten ermöglicht, durch globale Adressen auf Seiten des gemeinsam verwendeten Speicherplatzes zuzugreifen.

[0133] In den **Fig.** 9 und 10 ist ein Beispiel einer Verzeichnisstruktur dargestellt, die einen Zugriff auf den gemeinsam verwendeten Speicherplatz bereitstellt. **Fig.** 9 zeigt eine Verzeichnisseite **320**, die einen Seitenkopf **322** und Verzeichniseinträge **324** und **326** aufweist, wobei jeder Verzeichniseintrag ein Bereichsfeld **330**, ein Feld **332** des verantwortlichen Knotens und ein Adressenfeld **334** aufweist. Die Verzeichnisseiten können durch einen Verzeichnisseitengenerator erzeugt werden, der ein von der Verzeichnisverwaltungseinheit gesteuertes Softwaremodul sein kann. Es ist verständlich, daß die Verzeichnisverwaltungseinheit mehrere Verzeichnisse unter Einschluß eines Verzeichnisses für die globale Platte und eines Verzeichnisses für die globalen RAM-Verzeichnisse erzeugen kann. Die dargestellte Verzeichnisseite **320** kann eine Seite des globalen Adressenplatzes in der Art eines 4-Kilobyte-Abschnitts des gemeinsam verwendeten Adressenplatzes sein. Daher kann die Verzeichnisseite ebenso wie die anderen Seiten, für die die Verzeichnisseiten Zugang gewähren, in dem verteilten gemeinsam verwendeten Speicherplatz gespeichert werden.

[0134] Wie in **Fig.** 9 weiterhin dargestellt ist, weist jede Verzeichnisseite **120** einen Seitenkopf **322** auf, der Attributinformationen für diesen Seitenkopf einschließt, die typischerweise Metadaten für die Verzeichnisseite sind, und sie weist weiterhin Verzeichniseinträge in der Art der dargestellten Verzeichniseinträge **324** und **326** auf, die einen Index in einen Abschnitt des gemeinsam verwendeten Adressenraums bereitstellen, wobei dieser Abschnitt aus einer oder mehreren Seiten, einschließlich aller Seiten des verteilten gemeinsam verwendeten Speicherplatzes, bestehen kann. Die dargestellte Verzeichnisseite **320** weist Verzeichniseinträge auf, die einen ausgewählten Bereich globaler Adressen des gemeinsam verwendeten Speicherplatzes indexieren. Hierzu kann der Verzeichnisgenerator einen Bereichsgenerator aufweisen, so daß jeder Verzeichniseintrag ein Bereichsfeld **330** aufweisen kann, das den Anfang eines Adressenbereichs beschreibt, den dieser Eintrag lokalisiert.

[0135] Demgemäß kann jede Verzeichnisseite **320** mehrere Verzeichniseinträge, wie bspw. Einträge **324** und **326**, aufweisen, die den Adressenraum in einen Untersatz von Adressenbereichen unterteilen können. Beispielsweise weist die dargestellte Verzeichnisseite **320** zwei Verzeichniseinträge **324** und **326** auf. Die Verzeichniseinträge **324** und **326** können beispielsweise den Adressenraum in zwei Unterabschnitte unterteilen. In diesem Beispiel könnte der Anfangsadressenbereich des Verzeichniseintrags **324** die Basisadresse des Adressenraums sein, und der Anfangsadressenbereich des Verzeichniseintrags **326** könnte die Adresse für die obere Hälfte des Speicherplatzes sein. Dementsprechend liefert der Verzeichniseintrag **324** einen Index für Seiten, die in dem Adressenraum zwischen der Basisadresse und dem Mittelpunkt des Speicherplatzes gespeichert sind, und der Verzeichniseintrag **326** liefert komplementär dazu einen Index für Seiten, die in dem Adressenraum gespeichert sind, der von dem Mittelpunkt des Adressenraums bis zu der höchsten Adresse reicht.

[0136] **Fig.** 9 zeigt weiterhin eine Verzeichnisseite **320**, die in jedem Verzeichniseintrag ein verantwortliches Knotenfeld **332** und das globale Adressenfeld **334** der abhängigen Seite aufweist. Diese Felder **332**, **334** liefern weitere Ortsinformationen für die Daten, die auf Seiten innerhalb des im Feld **330** identifizierten Adressenbereichs gespeichert sind.

[0137] **Fig.** 10 zeigt ein Verzeichnis **340**, das aus Verzeichnisseiten ähnlich den in **Fig.** 9 dargestellten besteht. **Fig.** 10 zeigt, daß das Verzeichnis **340** Verzeichnisseiten **342**, **350**–**354** und **360**–**366** aufweist. **Fig.** 10 zeigt weiterhin, daß das Verzeichnis **340** Ortsinformationen für die in **Fig.** 10 als die Seiten **370**–**384** dargestellten Seiten des verteilten gemeinsam verwendeten Speicherplatzes bereitstellt.

[0138] Die in **Fig.** 10 dargestellte Verzeichnisseite **342** wirkt wie eine Wurzelverzeichnisseite und kann sich an einer statischen Adresse befinden, die jedem mit dem verteilten Adressenraum gekoppelten Knoten bekannt ist. Die Wurzelverzeichnisseite **342** weist drei Verzeichniseinträge **344**, **346** und **348** auf. Jeder in **Fig.** 10 dargestellte Verzeichniseintrag weist Verzeichniseinträge ähnlich den in **Fig.** 9 dargestellten auf. Beispielsweise weist der Verzeichniseintrag **344** eine Variable Co, die das Adressenbereichsfeld **330** repräsentiert, eine Variable Nj, die das Feld **332** repräsentiert, und eine Variable Cs, die das Feld **334** repräsentiert, auf. Die dargestellte Wurzelverzeichnisseite **342** unterteilt den Adressenraum in drei Bereiche, die als ein Adressenbereich, der sich zwischen den Adressen Co und Cd erstreckt, ein zweiter Adressenbereich, der sich zwischen den Adressen Cd und Cg erstreckt, und ein dritter Adressenbereich, der sich zwischen Cg und der höchsten Speicherstelle des Adressenraums erstreckt, dargestellt sind.

[0139] Wie in **Fig.** 10 weiter dargestellt ist, weist jeder Verzeichniseintrag **344**, **346** und **348** auf eine untergeordnete Verzeichnisseite, die als Verzeichnis-

seiten **350**, **352** und **354** dargestellt ist, von denen jede weiterhin den Adressenbereichsindex durch den zugeordneten Verzeichniseintrag des Wurzelverzeichnisses **342** unterteilt. In **Fig.** 9 wird dieser Unterteilungsprozeß fortgesetzt, weil jede der Verzeichnisseiten **350**, **352** und **354** jeweils wiederum Verzeichniseinträge aufweist, die untergeordnete Verzeichnisseiten einschließlich der dargestellten Beispiele der Verzeichnisseiten **360**, **362**, **364** und **366** lokalisieren.

[0140] In dem dargestellten Beispiel sind die Verzeichnisseiten **360**, **362**, **364** und **366** jeweils Zweigeinträge. Die Zweigeinträge enthalten Verzeichniseinträge, wie bspw. die Verzeichniseinträge **356** und **358** des Zweigeintrags **360**, die ein Bereichsfeld **330** und das verantwortliche Knotenfeld **332** speichern. Diese Zweigeinträge identifizieren eine Adresse und einen verantwortlichen Knoten für die Seite in dem verteilten Speicherplatz, auf die zugegriffen wird, wie bspw. die dargestellten Seiten **370**–**384**. Beispielsweise verweist der Zweigeintrag **356**, wie in **Fig.** 10 dargestellt ist, auf die Seite **370**, die dem Bereichsfeld **330** des Zweigeintrags **356** entspricht, die für einen Zweigeintrag die Seite ist, auf die zugegriffen wird. Auf diese Weise liefert die Verzeichnisstruktur **340** Ortsinformationen für in dem verteilten Adressenraum gespeicherte Seiten.

[0141] Bei der in **Fig.** 10 dargestellten Ausführungsform kann ein Knotenwähler einen verantwortlichen Knoten für jede Seite auswählen, wie vorstehend beschrieben wurde, so daß der Zweigeintrag **356** Informationen zu der Adresse und zu dem verantwortlichen Knoten der lokalisierten Seite bereitstellt. Dementsprechend verfolgt dieses Verzeichnis die Eigentümerschaft und die Verantwortlichkeit für Daten, um eine Indirektheitsebene zwischen dem Verzeichnis und dem physikalischen Ort der Daten bereitzustellen. Während eines Speicherzugriffsvorgangs übergibt das Speicheruntersystem **270** an den im Zweigeintrag **356** angegebenen verantwortlichen Knoten die Adresse der Seite, auf die zugegriffen wird. Das gemeinsam verwendete Speicheruntersystem dieses Knotens kann einen Knoten unter Einschluß des Eigentümerknotens identifizieren, der eine Kopie der Seite, auf die zugegriffen wird, speichert. Diese Identifikation eines Knotens, der eine Kopie aufweist, kann von dem RAM-Kopiersatz oder von dem Platten-Kopiersatz des verantwortlichen Knotens vorgenommen werden. Der Knoten mit einer in seinem lokalen physikalischen Speicher gespeicherten Kopie, wie bspw. der Eigentümerknoten, kann seine lokalen Cache-Elemente einschließlich des lokalen RAM-Cache-Speichers und des lokalen Platten-Cache-Speichers verwenden, um anhand des globalen Adressensignals einen physikalischen Ort der Daten zu identifizieren, die auf der Seite gespeichert sind, auf die zugegriffen wird. Das Cache-Element kann das Betriebssystem des Eigentümerknotens verwenden, um auf die Speichervorrichtung zuzugreifen, die diesen physikalischen Ort enthält, damit auf die Daten zugegriffen werden kann, die auf der Seite gespeichert sind. Für einen Speicherlesevorgang oder für einen anderen ähnlichen Vorgang können die aus dem physikalischen Speicher des Eigentümerknotens gelesenen Daten über das Netzwerk an das Speicheruntersystem des Knotens übergeben werden, der das Lesen anfordert, und nachfolgend in dem virtuellen Speicherplatz des anfordernden Knotens gespeichert werden, um von diesem Knoten verwendet zu werden.

[0142] Wiederum mit Bezug auf **Fig.** 10 ist ersichtlich, daß die dargestellte Verzeichnisstruktur **340** eine hierarchische Struktur aufweist. Hierzu liefert die Verzeichnisstruktur **340** eine Struktur, die den Speicherplatz kontinuierlich in immer kleinere Abschnitte unterteilt. Weiterhin ist jeder Abschnitt durch Verzeichnisseiten derselben Struktur, jedoch Indexadressenräume unterschiedlicher Größen, dargestellt. Wenn Seiten erzeugt oder gelöscht werden, fügt ein Linker die Seiten in das Verzeichnis ein oder löscht die Seiten aus diesem. Bei einer Ausführungsform ist der Linker ein Softwaremodul zum Verbinden von Datenstrukturen.

[0143] Der Linker kann ansprechend auf die Adressenbereiche arbeiten, um die dargestellte hierarchische Struktur bereitzustellen. Dementsprechend bietet das dargestellte Verzeichnis **340** ein skalierbares Verzeichnis für den gemeinsam verwendeten Adressenraum. Weiterhin werden die Verzeichnisseiten in dem verteilten Adressenraum gespeichert und von dem verteilten gemeinsam verwendeten Speichersystem unterhalten. Eine Wurzel für das Verzeichnis kann an bekannten Stellen gespeichert werden, um ein Urladen des Systems zu ermöglichen. Folglich werden häufig verwendete Seiten kopiert und verteilt und selten verwendete Seiten von der Platte geschoben. In ähnlicher Weise migrieren Verzeichnisseiten zu den Knoten, die am häufigsten auf sie zugreifen, wodurch ein Grad der Selbstorganisation bereitgestellt wird, der den Netzwerkverkehr verringert.

[0144] **Fig.** 11 zeigt das Verzeichnis aus **Fig.** 10, das von einem erfindungsgemäßen System verwendet wird. Insbesondere zeigt **Fig.** 11 ein System **400**, das zwei Knoten **406a** und **406b**, eine Verzeichnisstruktur **340** und ein Paar lokaler Speicher mit flüchtigen Speichervorrichtungen **264a** und **264b** und permanenten Speichervorrichtungen **262a** und **262b** aufweist. Ein dargestellter Knoten **406a** weist einen Adressenkunden **408a**, eine globale Adresse **410a** sowie eine Schnittstelle **242a**, eine Verzeichnisverwaltungseinheit **244a** und eine Speichersteuereinrichtung **246a** auf. Ein Knoten **406b** weist entsprechende Elemente auf. Die Knoten sind durch das Netzwerk **254** verbunden. Das Verzeichnis **340** weist eine Wurzelseite, Verzeichnisseiten A–F und Seiten 1–5 auf.

[0145] Jeder Knoten **406a** und **406b** arbeitet so, wie vorstehend erörtert wurde. Die dargestellten Adressenkunden **408a** und **408b** können ein Anwendungsprogramm, ein Dateisystem, eine Hardwarevorrichtung oder ein anderes solches Element sein, das den

Zugriff auf den virtuellen Speicher anfordert. Beim Betrieb fordern die Adressenkunden **408a** und **408b** eine Adresse oder einen Adressenbereich an, und die Verzeichnisverwaltungseinheit kann einen globalen Adressengenerator aufweisen, der dem Kunden die angeforderte Adresse oder einen Zeiger auf die angeforderte Adresse liefert. Wenn Adressen erzeugt werden, erzeugen die jeweiligen Verzeichnisverwaltungseinheiten **244a** und **244b** Verzeichnisseiten und speichern die Seiten in der Verzeichnisstruktur **340**. Wie dargestellt ist, verfolgt die Verzeichnisstruktur **340** die Abschnitte des Adressenraums, die von dem System **400** verwendet werden, und der physikalische Speicher für jede Seite ist innerhalb der lokalen Speicher bereitgestellt.

[0146] Wie in **Fig.** 11 dargestellt ist, werden die den Verzeichnisseiten zugeordneten Daten verteilt über die zwei lokalen Speicher gespeichert, und es können Vervielfältigungskopien existieren. Wie vorstehend beschrieben wurde und nun in **Fig.** 11 dargestellt ist, können die Daten oder die Seite zwischen verschiedenen lokalen Speichern bewegt werden und auch zwischen dem flüchtigen und dem permanenten Speicher bewegt werden. Die Datenbewegung kann ansprechend auf Datenanforderungen, die von Speicherbenutzern in der Art von Anwendungsprogrammen gemacht werden, oder unter Verwendung der vorstehend beschriebenen Migrationssteuereinrichtung erfolgen. Wie vorstehend auch beschrieben wurde, kann die Bewegung von Daten zwischen verschiedenen Speicherstellen auftreten, ohne daß Änderungen an dem Verzeichnis **340** erforderlich wären. Dies wird erreicht, indem ein Verzeichnis **340** bereitgestellt wird, das von der physikalischen Stelle der Daten entkoppelt ist, indem ein Zeiger auf einen verantwortlichen Knoten verwendet wird, der die Datenspeicherstelle verfolgt. Wenngleich sich dementsprechend die Datenspeicherstelle ändern kann, kann der verantwortliche Knoten konstant bleiben, wodurch es überflüssig wird, das Verzeichnis **340** zu wechseln.

[0147] Durchschnittsfachleuten werden Abänderungen, Modifikationen und andere Implementationen von dem einfallen, was hier beschrieben wird, ohne von dem Gedanken und von dem Schutzumfang der beanspruchten Erfindung abzuweichen. Dementsprechend soll die Erfindung nicht durch die vorstehende erläuternde Beschreibung sondern vielmehr durch den Schutzumfang der folgenden Ansprüche definiert sein.

Patentansprüche

1. Verfahren zum Bereitstellen einer verteilten Steuerung über einen strukturierten Datenspeicher, mit den folgenden Schritten:

Bereitstellen einer Mehrzahl von durch ein Netzwerk miteinander verbundenen Knoten, wobei jeder der Mehrzahl von Knoten einen gemeinsam verwendeten adressierbaren Speicherplatz eines gemeinsam verwendeten Speichersystems geteilt verwendet und (i) eine Schnittstelle zum Zugreifen auf das Netzwerk, (ii) eine lokale flüchtige Speichervorrichtung, die mit dem Knoten gekoppelt ist und einen flüchtigen Speicher bereitstellt, (iii) eine lokale permanente Speichervorrichtung, die mit dem Knoten gekoppelt ist und einen permanenten Speicher bereitstellt, und (iv) ein gemeinsam verwendetes Speicheruntersystem zum Abbilden eines Abschnitts des gemeinsam verwendeten adressierbaren Speicherplatzes in mindestens einen Abschnitt des permanenten und des flüchtigen Speichers, um dadurch einen adressierbaren permanenten und flüchtigen Speicher bereitzustellen, der von jedem der Mehrzahl von Knoten ansteuerbar ist, aufweist, wobei das gemeinsam verwendete Speicheruntersystem (a) einen Verteiler zum Abbilden von Abschnitten des adressierbaren Speicherplatzes über die Mehrzahl von lokalen permanenten und flüchtigen Speichervorrichtungen, um den adressierbaren Speicherplatz über die Mehrzahl von lokalen permanenten und flüchtigen Speichervorrichtungen zu verteilen, und (b) eine Verzeichnisverwaltungseinheit zum Verfolgen der abgebildeten Abschnitte des adressierbaren Speicherplatzes, um Informationen bereitzustellen, die angeben, welche Abschnitte des adressierbaren Speicherplatzes auf welche der lokalen permanenten und flüchtigen Speichervorrichtungen abgebildet sind, aufweist,

Speichern einer Ausprägung eines Datensteuerprogramms zum Manipulieren des strukturierten Datenspeichers an jedem Knoten, um mehrere verteilte Ausprägungen des Datensteuerprogramms bereitzustellen,

Verknüpfen jeder Ausprägung des Datensteuerprogramms mit dem gemeinsam verwendeten Speichersystem und

Betreiben jeder Ausprägung des Datensteuerprogramms, um das gemeinsam verwendete Speichersystem als eine Speichervorrichtung zu verwenden, in der der strukturierte Datenspeicher enthalten ist, wobei die Koordinaten des gemeinsam verwendeten Speichersystems auf den strukturierten Datenspeicher zugreifen, um eine verteilte Steuerung des strukturierten Datenspeichers bereitzustellen.

2. Verfahren nach Anspruch 1, wobei der Verknüpfungsschritt die weiteren Schritte aufweist:

Anweisen des Datensteuerprogramms, einen Strom im strukturierten Datenspeicher zu speichernder Daten bereitzustellen, und

Anweisen des Datensteuerprogramms, das gemeinsam verwendete Speichersystem als eine Speichervorrichtung mit einem einzigen Knoten zu betreiben.

3. Verfahren nach Anspruch 1, wobei der strukturierte Datenspeicher ein Dateisystem aufweist und wobei das Datensteuerprogramm ein Dateisteuerprogramm zum Manipulieren des Dateisystems aufweist, wobei das gemeinsam verwendete Speichersystem den Zugriff auf das Dateisystem steuert, um

ein gemeinsam verwendetes Dateisystem bereitzustellen.

4. Verfahren nach Anspruch 3, bei dem des weiteren
das gemeinsam verwendete Dateisystem mit einem Dateiverzeichnis versehen wird und
das gemeinsam verwendete Speichersystem so betrieben wird, daß das Dateiverzeichnis innerhalb eines gemeinsam verwendeten Speicherplatzes gehalten wird.

5. Verfahren nach Anspruch 4, bei dem des weiteren das Dateiverzeichnis als eine Mehrzahl von innerhalb des gemeinsam verwendeten Speicherplatzes gespeicherten logischen Dateipartitionen organisiert werden.

6. Verfahren nach Anspruch 4, das des weiteren den Schritt des Koordinierens des geteilten Zugriffs auf Daten innerhalb des strukturierten Speichers durch Sperren innerhalb eines gemeinsam verwendeten Speicherplatzes gespeicherter Verzeichnisse umfaßt.

7. Verfahren nach Anspruch 3, bei dem des weiteren für eine innerhalb des gemeinsam verwendeten Dateisystems gespeicherte Datei ein Dateideskriptor erzeugt wird, der einen Speicher für einen Bezeichner aufweist, der einen Abschnitt eines gemeinsam verwendeten Speicherplatzes repräsentiert.

8. Verfahren nach Anspruch 7, bei dem des weiteren benachbarte Abschnitte des gemeinsam verwendeten Speicherplatzes zugeordnet werden, die jeweils durch einen jeweiligen Bezeichner repräsentiert sind, um reduzierte Buchhaltungsinformationen für die Datei bereitzustellen.

9. Verfahren nach Anspruch 7, bei dem des weiteren benachbarte Segmente einer Speichervorrichtung zum Speichern von Daten, die den benachbarten Abschnitten des gemeinsam verwendeten Speicherplatzes zugeordnet sind, reserviert werden, um den Zugriff auf den physikalischen Speicher für die Datei zu optimieren.

10. Verfahren nach Anspruch 1, wobei der strukturierte Datenspeicher ein Datenbanksystem aufweist und wobei das Datensteuerprogramm ein Datenbank-Steuerprogramm zum Manipulieren des Datenbanksystems aufweist, wobei das gemeinsam verwendete Speichersystem den Zugriff auf das Datenbanksystem steuert, um ein gemeinsam verwendetes Datenbanksystem bereitzustellen.

11. Verfahren nach Anspruch 10, bei dem des weiteren
das gemeinsam verwendete Datenbanksystem mit einem Datenbankverzeichnis und einem Satz von Indexstrukturen versehen wird, und
das gemeinsam verwendete Speichersystem betrieben wird, um das Datenbankverzeichnis und den Satz von Indexstrukturen innerhalb eines gemeinsam verwendeten Speicherplatzes zu erhalten.

12. Verfahren nach Anspruch 11, bei dem des weiteren das Datenbankverzeichnis als eine Mehrzahl von innerhalb des gemeinsam verwendeten Speicherplatzes gespeicherten Sätzen organisiert wird.

13. Verfahren nach Anspruch 10, das des weiteren die folgenden Schritte umfaßt:
Zuordnen von Zeitgleichzugriffs-Steuerstrukturen zu Abschnitten des Datenbanksystems,
Speichern der Zeitgleichzugriffs-Steuerstrukturen in dem gemeinsam verwendeten Speicherplatz und
Koordinieren des geteilten Zugriffs auf das Datenbanksystem durch Sperren von Zeitgleichzugriffs-Steuerstrukturen.

14. Verfahren nach Anspruch 13, das des weiteren das Sperren von Datenbankindizes umfaßt.

15. Verfahren nach Anspruch 13, das des weiteren das Sperren von Datenbankschlüsseln umfaßt.

16. Verfahren nach Anspruch 10, bei dem des weiteren für ein innerhalb des gemeinsam verwendeten Datenbanksystems gespeichertes Datenbankobjekt ein Datenbank-Datensatzdeskriptor erzeugt wird, der einen Speicher für einen Bezeichner aufweist, der einen Abschnitt eines gemeinsam verwendeten Speicherplatzes repräsentiert.

17. Verfahren nach Anspruch 16, bei dem des weiteren benachbarte Abschnitte des gemeinsam verwendeten Speicherplatzes zugeordnet werden, die jeweils durch einen jeweiligen Bezeichner repräsentiert sind, um reduzierte Buchhaltungsinformationen für den jeweiligen Datenbank-Datensatz bereitzustellen.

18. Verfahren nach Anspruch 16, bei dem des weiteren benachbarte Segmente einer Speichervorrichtung zum Speichern von Daten, die den benachbarten Abschnitten des gemeinsam verwendeten Speicherplatzes zugeordnet sind, reserviert werden, um den Zugriff auf den physikalischen Speicher für den Datenbank-Datensatz zu optimieren.

19. Verfahren nach Anspruch 1, wobei der strukturierte Datenspeicher ein Web-Serversystem umfaßt und wobei das Datensteuerprogramm ein Steuerprogramm zum Manipulieren des Web-Serversystems umfaßt und der Zugriff auf das Web-Serversystem gesteuert wird, um ein gemeinsam verwendetes Web-Serversystem bereitzustellen.

20. Verfahren nach Anspruch 19, bei dem des weiteren
das gemeinsam verwendete Web-Serversystem mit einem Verzeichnis versehen wird, das die Dateien ihrem Inhalt zuordnet und
das gemeinsam verwendete Speichersystem betrieben wird, um das Web-Serververzeichnis innerhalb eines gemeinsam verwendeten Speicherplatzes zu erhalten.

21. Verfahren nach Anspruch 19, bei dem des weiteren für eine innerhalb des gemeinsam verwendeten Web-Serversystems gespeicherte Datei ein Dateideskriptor erzeugt wird, der einen Speicher für einen Bezeichner aufweist, der einen Abschnitt eines gemeinsam verwendeten Speicherplatzes repräsentiert.

22. Verfahren nach Anspruch 21, bei dem des weiteren benachbarte Abschnitte des gemeinsam verwendeten Speicherplatzes zugeordnet werden, die jeweils durch einen jeweiligen Bezeichner repräsentiert sind, um reduzierte Buchhaltungsinformationen für die Dateien bereitzustellen.

23. Verfahren nach Anspruch 21, bei dem des weiteren benachbarte Segmente einer Speichervorrichtung zum Speichern von Daten, die den benachbarten Abschnitten des gemeinsam verwendeten Speicherplatzes zugeordnet sind, reserviert werden, um den Zugriff auf den physikalischen Speicher für die Dateien zu optimieren.

24. Verfahren nach Anspruch 1, bei dem des weiteren das gemeinsam verwendete Speichersystem betrieben wird, um gespeicherte Daten kohärent zu replizieren und so einen redundanten Datenspeicher bereitzustellen.

25. Verfahren nach Anspruch 24, bei dem des weiteren die kohärent replizierten Daten innerhalb verschiedener Speichervorrichtungen des Netzwerks gespeichert werden, um einen fehlertoleranten Betrieb bereitzustellen.

26. Verfahren nach Anspruch 1, bei dem des weiteren
Zeitgleichzugriffs-Steuerstrukturen Abschnitten des gemeinsam verwendeten Speicherplatzes zugeordnet werden,
die Zeitgleichzugriffs-Steuerstrukturen im gemeinsam verwendeten Speicherplatz gespeichert werden und
der geteilte Zugriff auf Daten innerhalb des strukturierten Speichers durch Sperren von Zeitgleichzugriffs-Steuerstrukturen koordiniert wird.

27. Verfahren nach Anspruch 26, bei dem des weiteren
eine Sperrobjekt-Datenstruktur mit Informationen erzeugt wird, die einen Sperrstatus auf Abschnitten des gemeinsam verwendeten Speicherplatzes repräsentieren, und
das Sperrobjekt innerhalb des gemeinsam verwendeten Speicherplatzes gespeichert wird, um dadurch eine gemeinsam verwendete Systemsperre bereitzustellen.

28. Verfahren nach Anspruch 26, wobei bei dem Sperrschritt der gemeinsam verwendete Speicher angewiesen wird, Bytebereichssperren zu erzeugen, die Sperren repräsentieren, die auf Abschnitte des gemeinsam verwendeten Speicherplatzes gesetzt sind.

29. Verfahren nach Anspruch 1, bei dem des weiteren jede Ausprägung des Datensteuerprogramms betrieben wird, um das gemeinsam verwendete Speichersystem als einen in Clustern angeordneten strukturierten Speicher zu verwenden, wobei das Speichersystem den Zugriff auf den in Clustern angeordneten strukturierten Speicher koordiniert, um eine verteilte Steuerung über den in Clustern angeordneten strukturierten Speicher bereitzustellen.

30. Verfahren zum Bereitstellen einer verteilten Steuerung über einen strukturierten Datenspeicher, mit folgenden Schritten:
Bereitstellen einer Mehrzahl von durch ein Netzwerk miteinander verbundener Knoten,
Speichern einer Ausprägung eines Datensteuerprogramms an jedem Knoten, um mehrere verteilte Ausprägungen des Datensteuerprogramms bereitzustellen, wobei das Datensteuerprogramm den Zugriff auf den strukturierten Datenspeicher manipuliert und steuert,
Verknüpfen jeder Ausprägung des Datensteuerprogramms mit einem gemeinsam verwendeten Speichersystem, das einen adressierbaren permanenten Datenspeicher bereitstellt,
Betreiben jeder Ausprägung des Datensteuerprogramms, um das gemeinsam verwendete Speichersystem als eine Speichervorrichtung zu verwenden, in der der strukturierte Datenspeicher enthalten ist, wobei die Koordinaten des gemeinsam verwendeten Speichersystems auf den strukturierten Datenspeicher zugreifen, um eine verteilte Steuerung des strukturierten Datenspeichers bereitzustellen,
Versehen der strukturierten Datenspeicher mit einem Verzeichnis, das die Dateien ihrem Inhalt zuordnet, und
Betreiben des gemeinsam verwendeten Speichersystems, um das Verzeichnis innerhalb eines gemeinsam verwendeten Speicherplatzes zu erhalten.

31. Verfahren nach Anspruch 30, bei dem der strukturierte Datenspeicher ein Web-Serversystem aufweist und das Datensteuerprogramm das Web-Serversystem manipuliert und den Zugriff auf das Web-Serversystem steuert, um ein gemeinsam

verwendetes Web-Serversystem bereitzustellen.

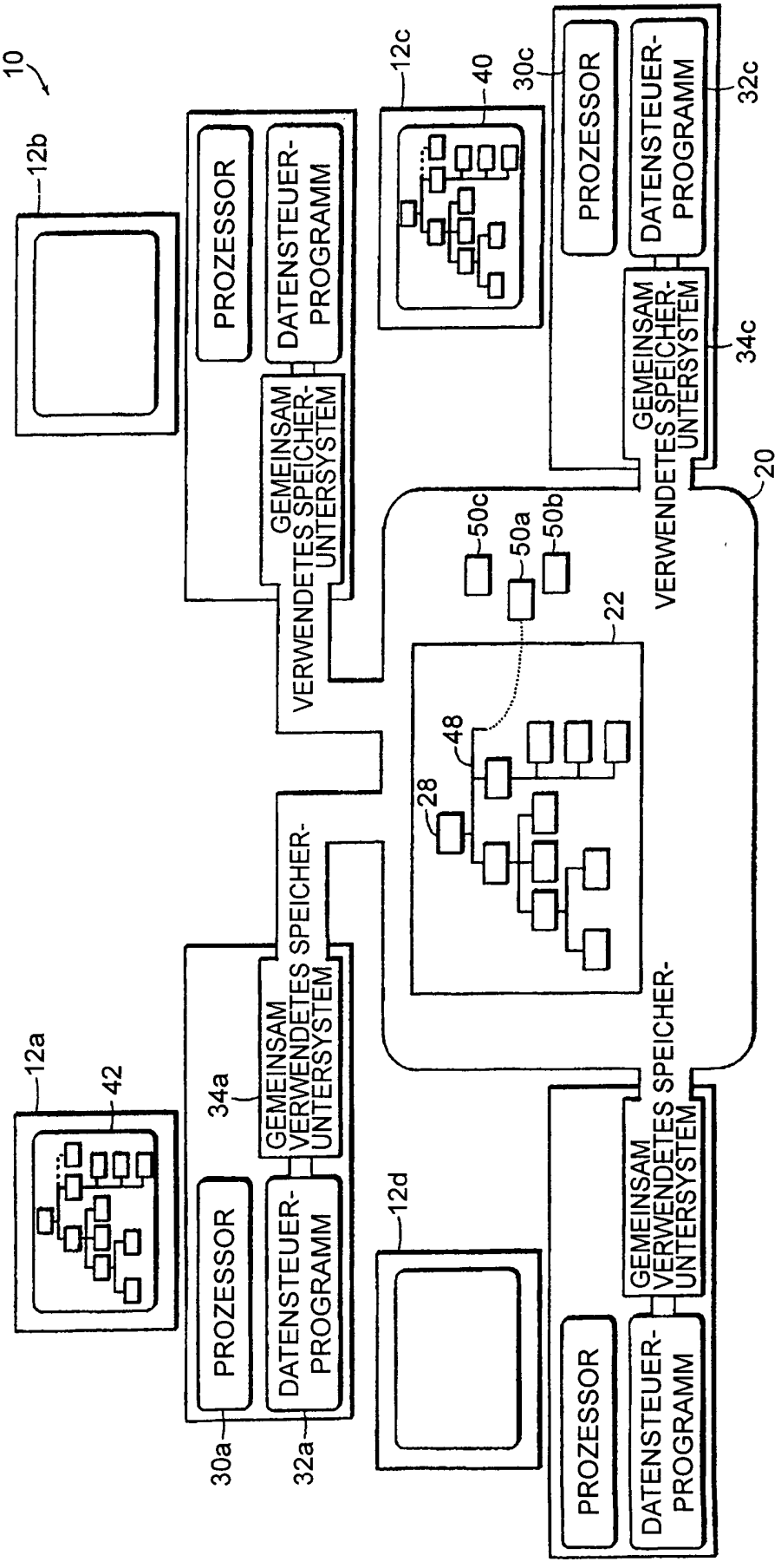
32. Verfahren nach Anspruch 31, bei dem des weiteren für eine innerhalb des gemeinsam verwendeten Web-Serversystems gespeicherte Datei ein Dateideskriptor mit einem Speicher für einen Bezeichner erzeugt wird, der einen Abschnitt eines gemeinsam verwendeten Speicherplatzes repräsentiert.

33. Verfahren nach Anspruch 31, bei dem des weiteren benachbarte Abschnitte des gemeinsam verwendeten Speicherplatzes zugeordnet werden, die jeweils durch einen jeweiligen Bezeichner repräsentiert sind, um, reduzierte Buchhaltungsinformationen für die Dateien bereitzustellen.

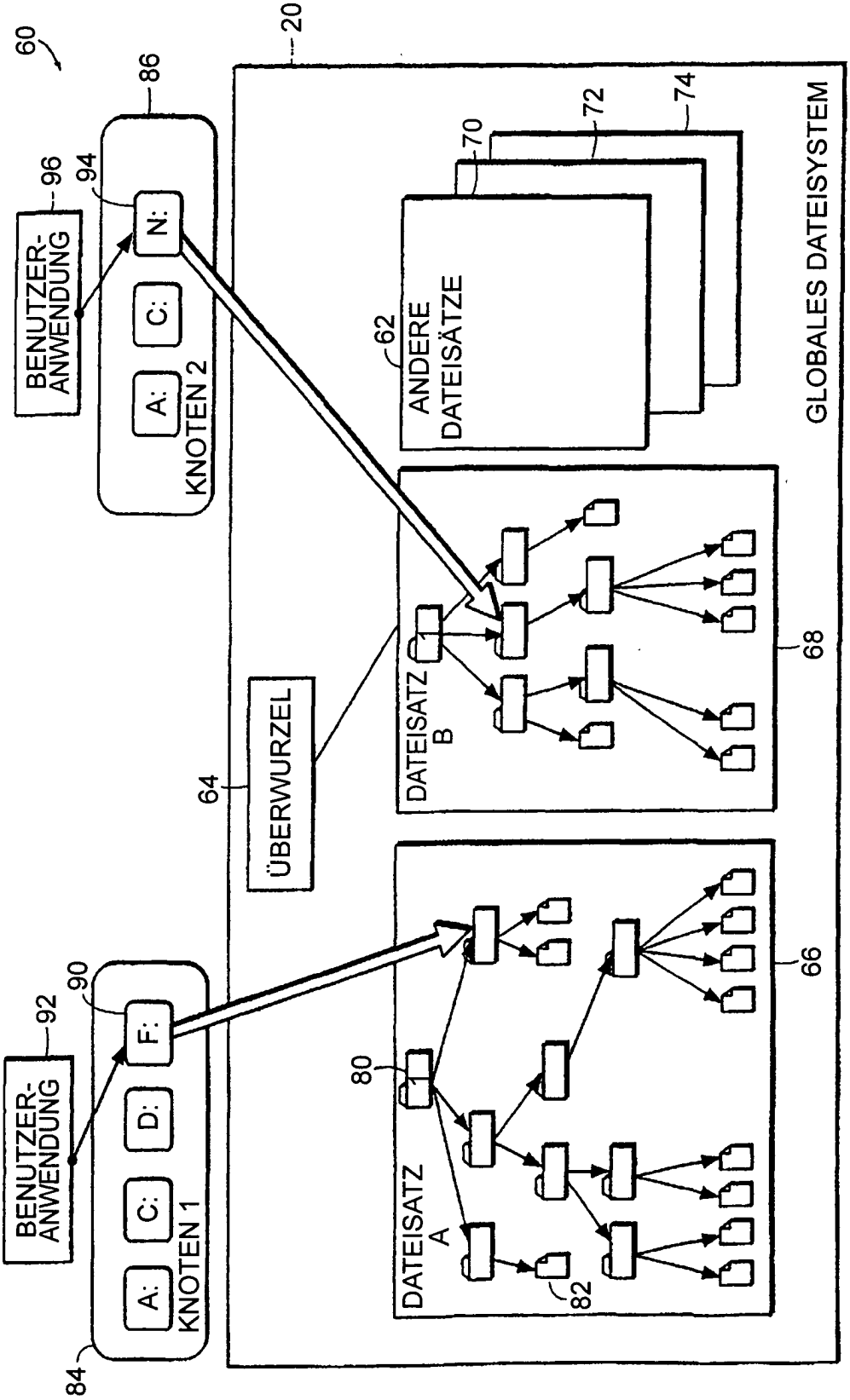
34. Verfahren nach Anspruch 31, bei dem des weiteren benachbarte Segmente einer Speichervorrichtung zum Speichern von Daten, die den benachbarten Abschnitten des gemeinsam verwendeten Speicherplatzes zugeordnet sind, reserviert werden, um den Zugriff auf den physikalischen Speicher für die Dateien zu optimieren.

Es folgen 11 Blatt Zeichnungen

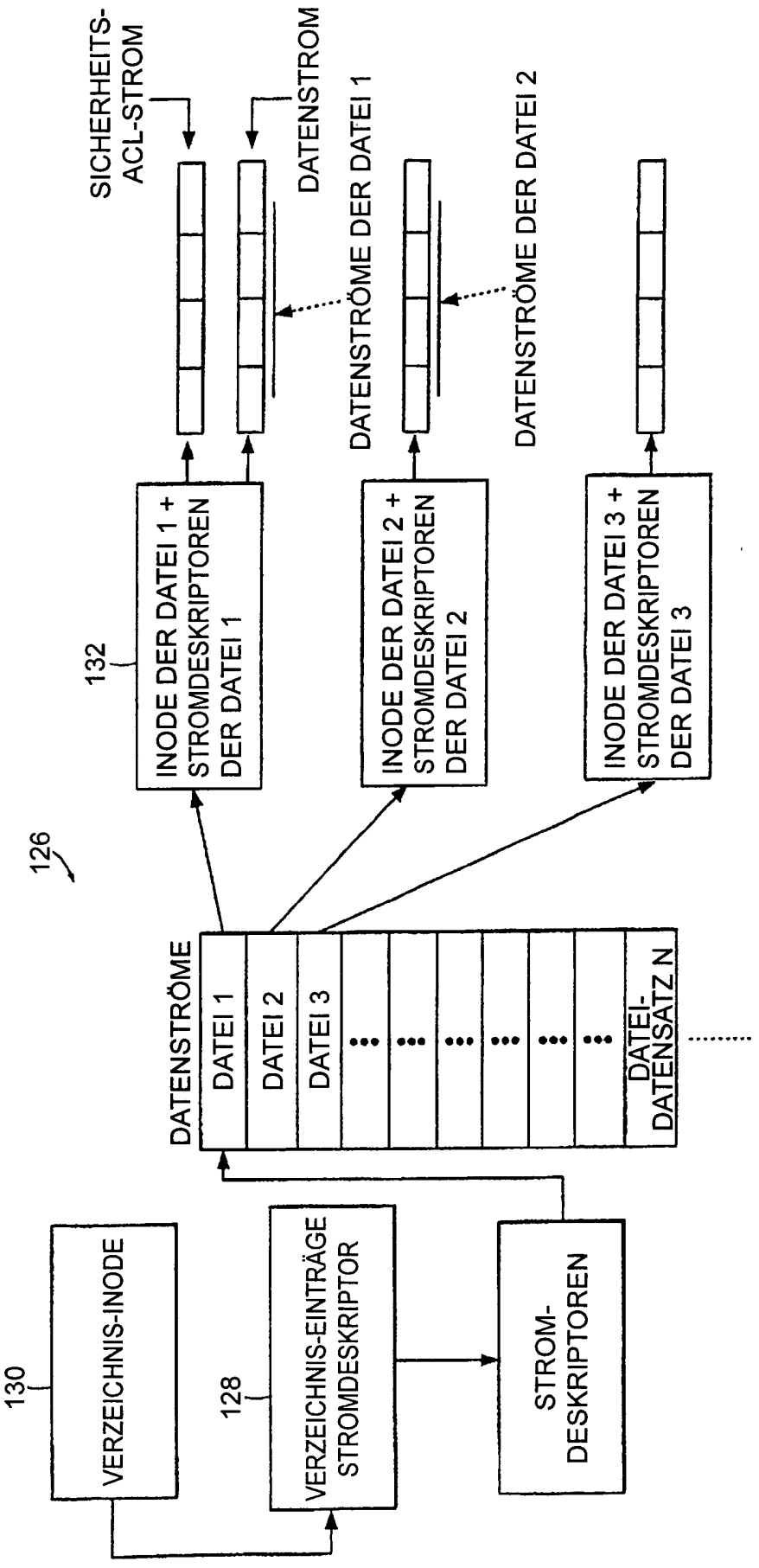
Anhängende Zeichnungen



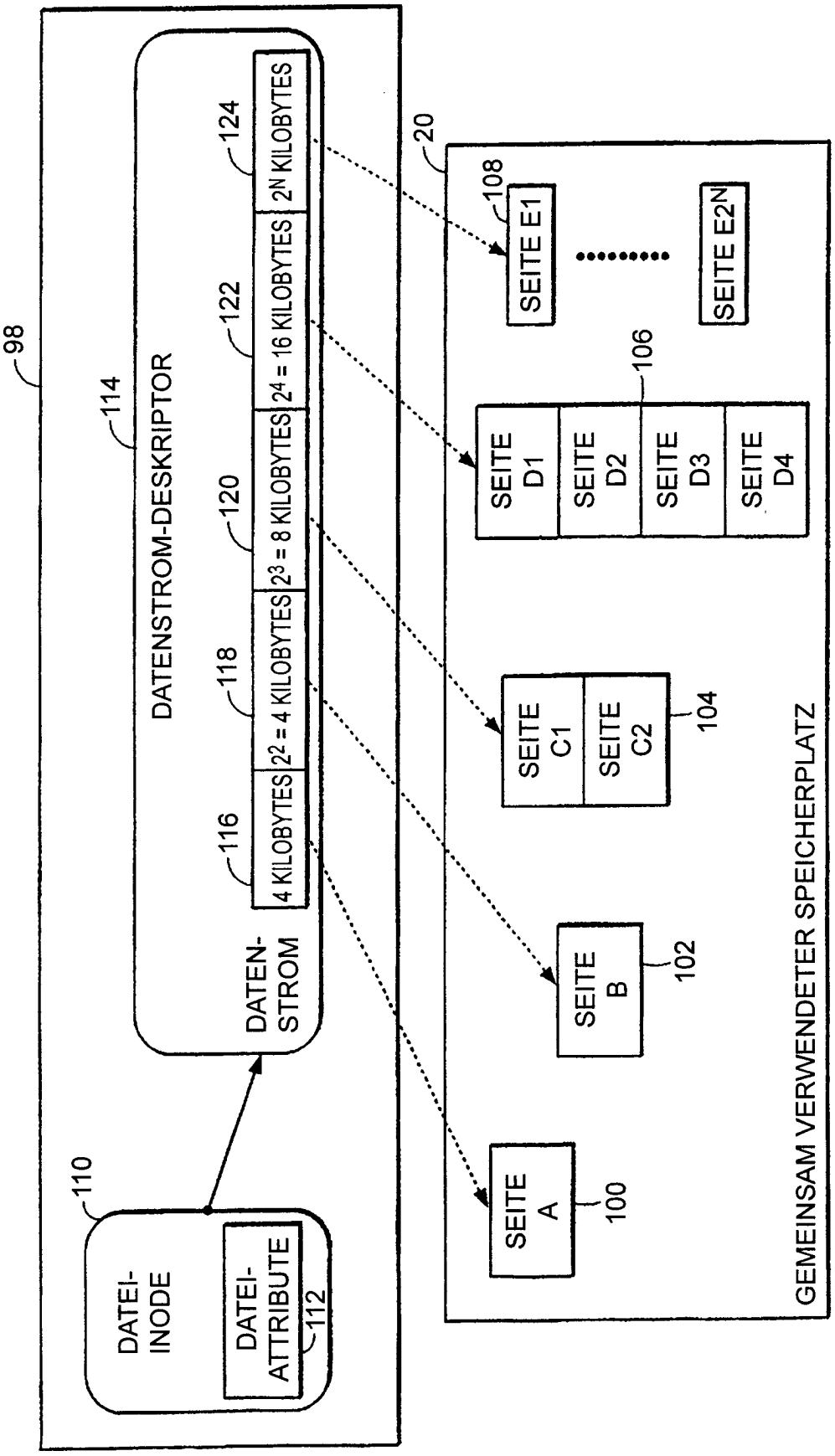
Figur 1


60
20
86
96
94
BENUTZER-ANWENDUNG
N:
C:
A:
KNOTEN 2
70
72
74
62
ANDERE DATEISÄTZE
GLOBALES DATEISYSTEM
DATEISATZ B
68
64
ÜBERWURZEL
92
90
BENUTZER-ANWENDUNG
F:
D:
C:
A:
KNOTEN 1
80
DATEISATZ A
66
82
84

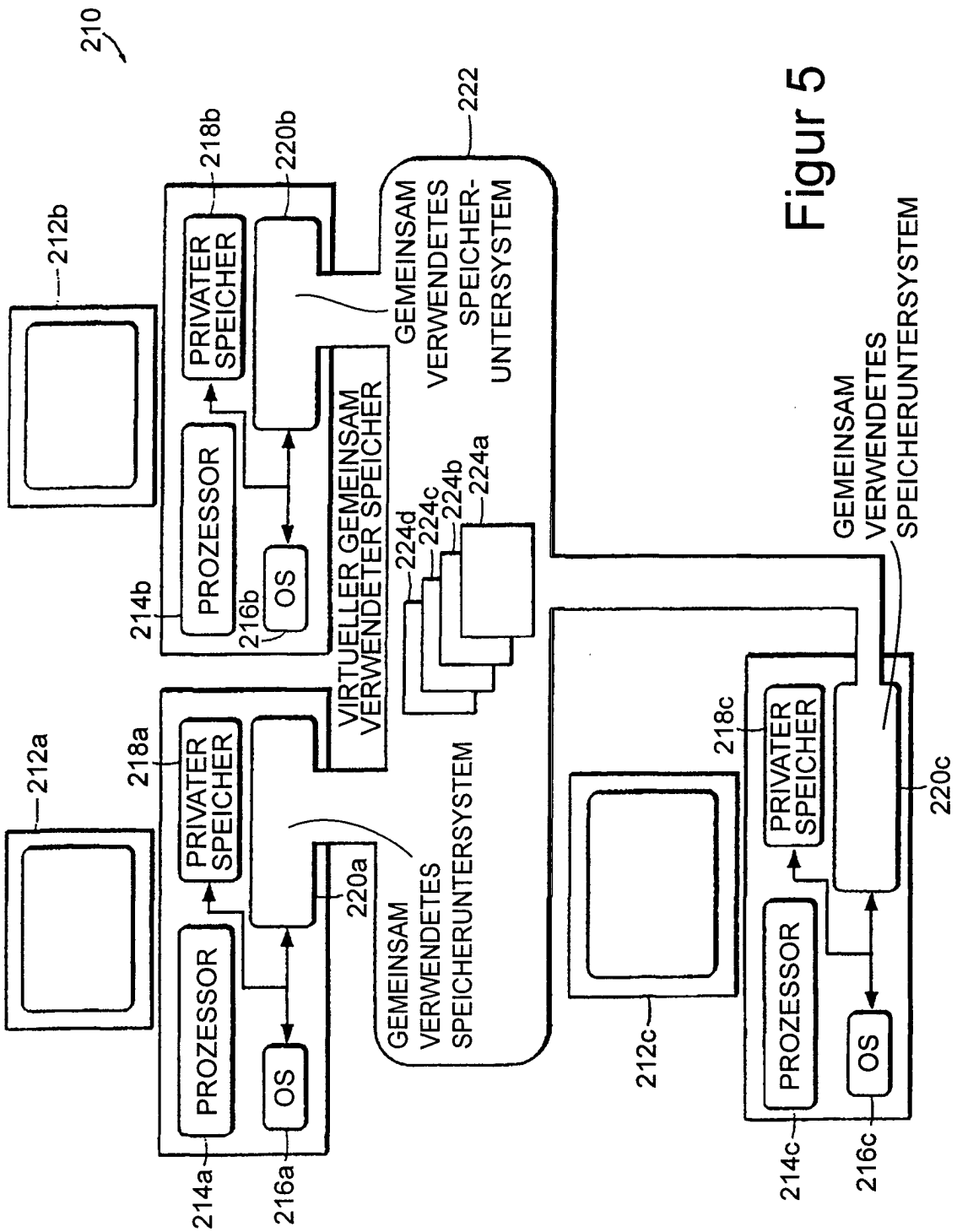
Figur 2


SICHERHEITS-
ACL-STROM
DATENSTROM
DATENSTRÖME DER DATEI 1
DATENSTRÖME DER DATEI 2
132
INODE DER DATEI 1 +
STROMDESKRIPTOREN
DER DATEI 1
INODE DER DATEI 2 +
STROMDESKRIPTOREN
DER DATEI 2
INODE DER DATEI 3 +
STROMDESKRIPTOREN
DER DATEI 3
126
DATENSTRÖME
DATEI 1
DATEI 2
DATEI 3
DATEI-
DATENSATZ N
130
VERZEICHNIS-INODE
128
VERZEICHNIS-EINTRÄGE
STROMDESKRIPTOR
STROM-
DESKRIPTOREN

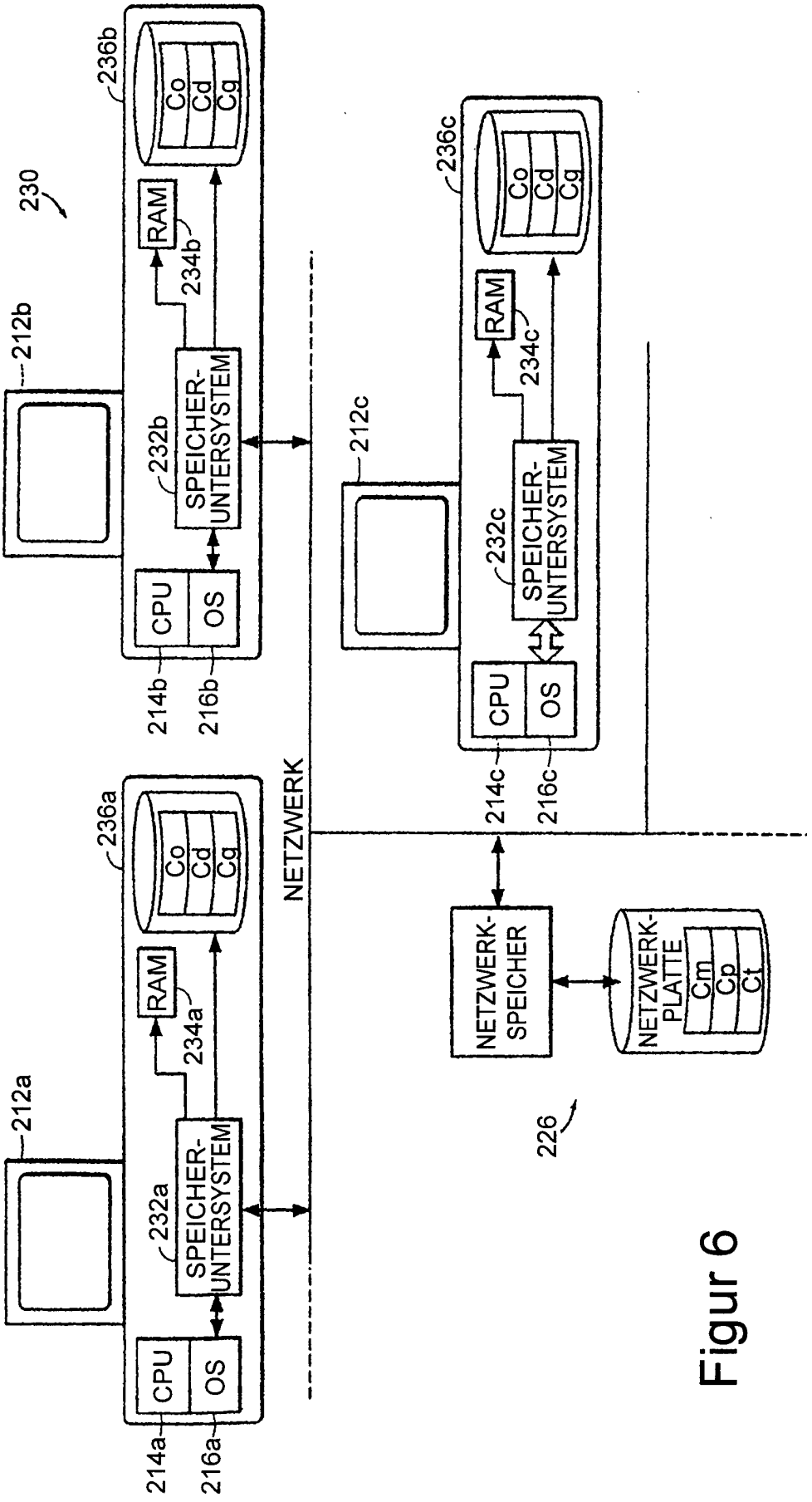
Figur 3


98
114
DATENSTROM-DESKRIPTOR
DATEN-STROM
116
4 KILOBYTES
118
2^2 = 4 KILOBYTES
120
2^3 = 8 KILOBYTES
122
2^4 = 16 KILOBYTES
124
2^N KILOBYTES
110
DATEI-INODE
DATEI-ATTRIBUTE
112
20
GEMEINSAM VERWENDETER SPEICHERPLATZ
SEITE A
100
SEITE B
102
SEITE C1
SEITE C2
104
SEITE D1
SEITE D2
SEITE D3
SEITE D4
106
108
SEITE E1
SEITE E2^N

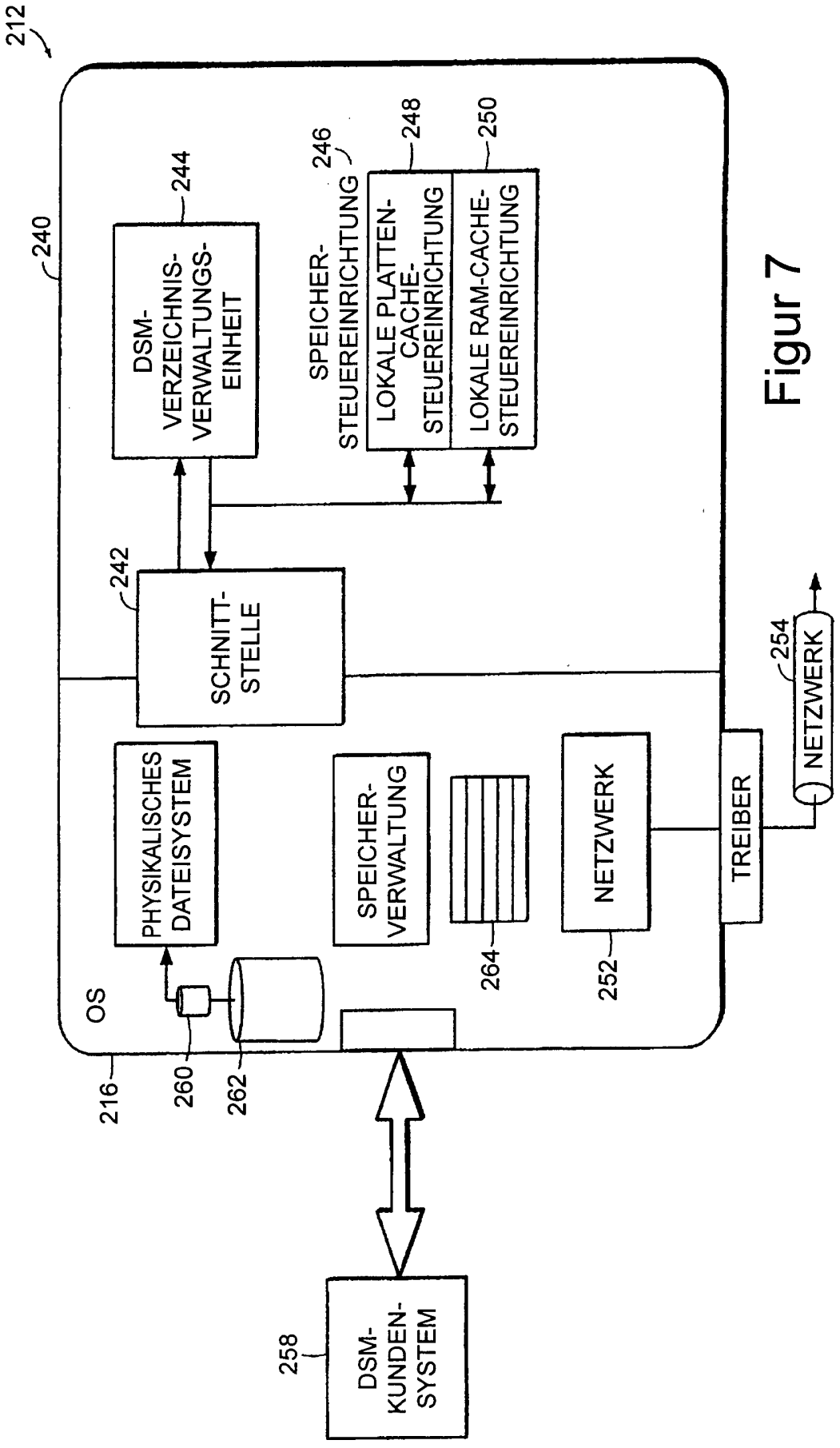
Figur 4


210
212a
212b
212c
218a
218b
218c
PRIVATER SPEICHER
PROZESSOR
OS
214a
214b
214c
216a
216b
216c
220a
220b
220c
VIRTUELLER GEMEINSAM VERWENDETER SPEICHER
GEMEINSAM VERWENDETES SPEICHER-UNTERSYSTEM
GEMEINSAM VERWENDETES SPEICHERUNTERSYSTEM
222
224a
224b
224c
224d

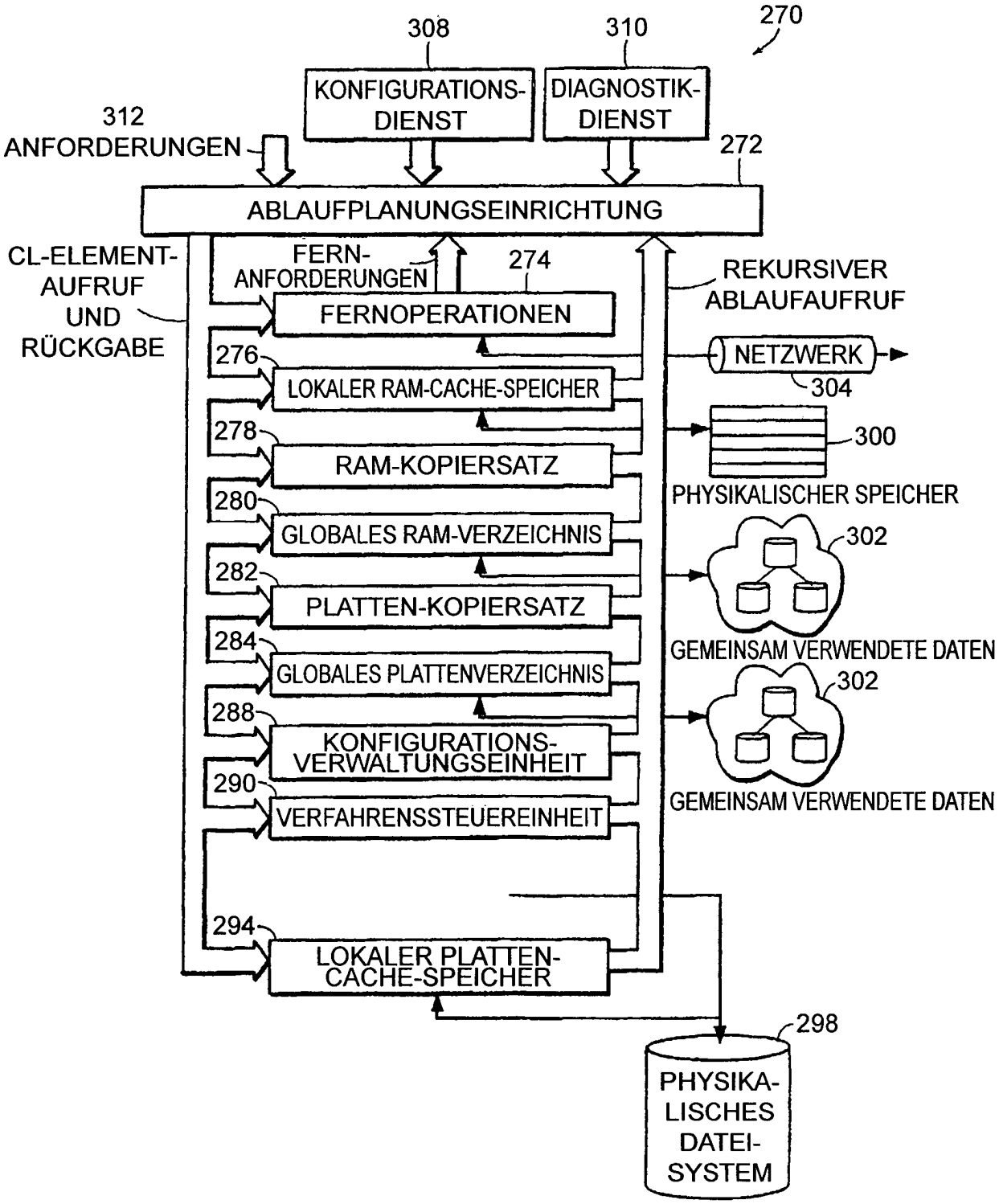
Figur 5


230
212a
212b
212c
232a
232b
232c
SPEICHER-UNTERSYSTEM
RAM
234a
234b
234c
CPU
OS
214a
216a
214b
216b
214c
216c
236a
236b
236c
Co
Cd
Cg
NETZWERK
NETZWERK-SPEICHER
NETZWERK-PLATTE
Cm
Cp
Ct
226

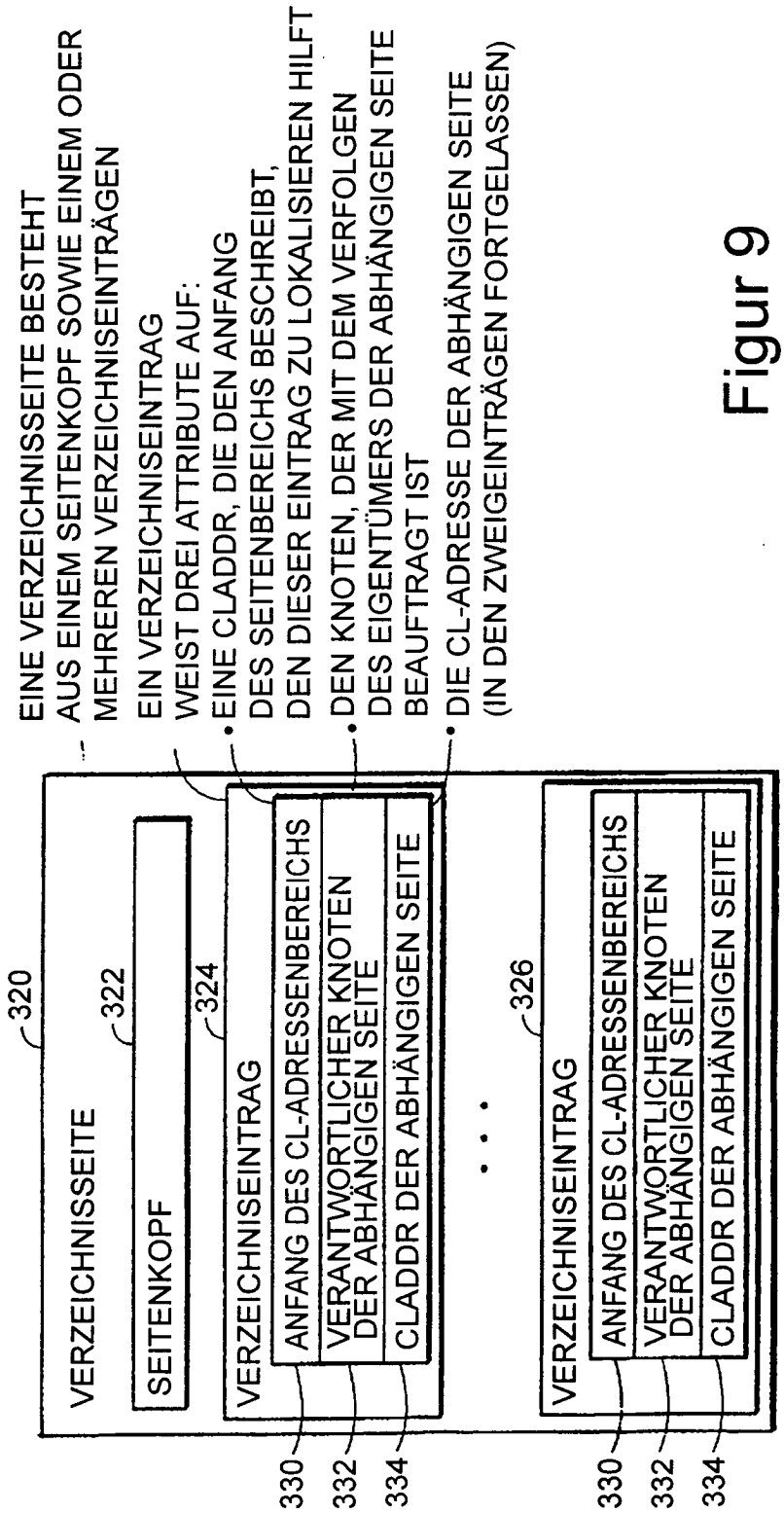
Figur 6


212
240
DSM-VERZEICHNIS-VERWALTUNGS-EINHEIT
244
SPEICHER-STEUEREINRICHTUNG
246
LOKALE PLATTEN-CACHE-STEUEREINRICHTUNG
248
LOKALE RAM-CACHE-STEUEREINRICHTUNG
250
SCHNITT-STELLE
242
PHYSIKALISCHES DATEISYSTEM
SPEICHER-VERWALTUNG
264
NETZWERK
252
TREIBER
254
NETZWERK
OS
216
260
262
DSM-KUNDEN-SYSTEM
258

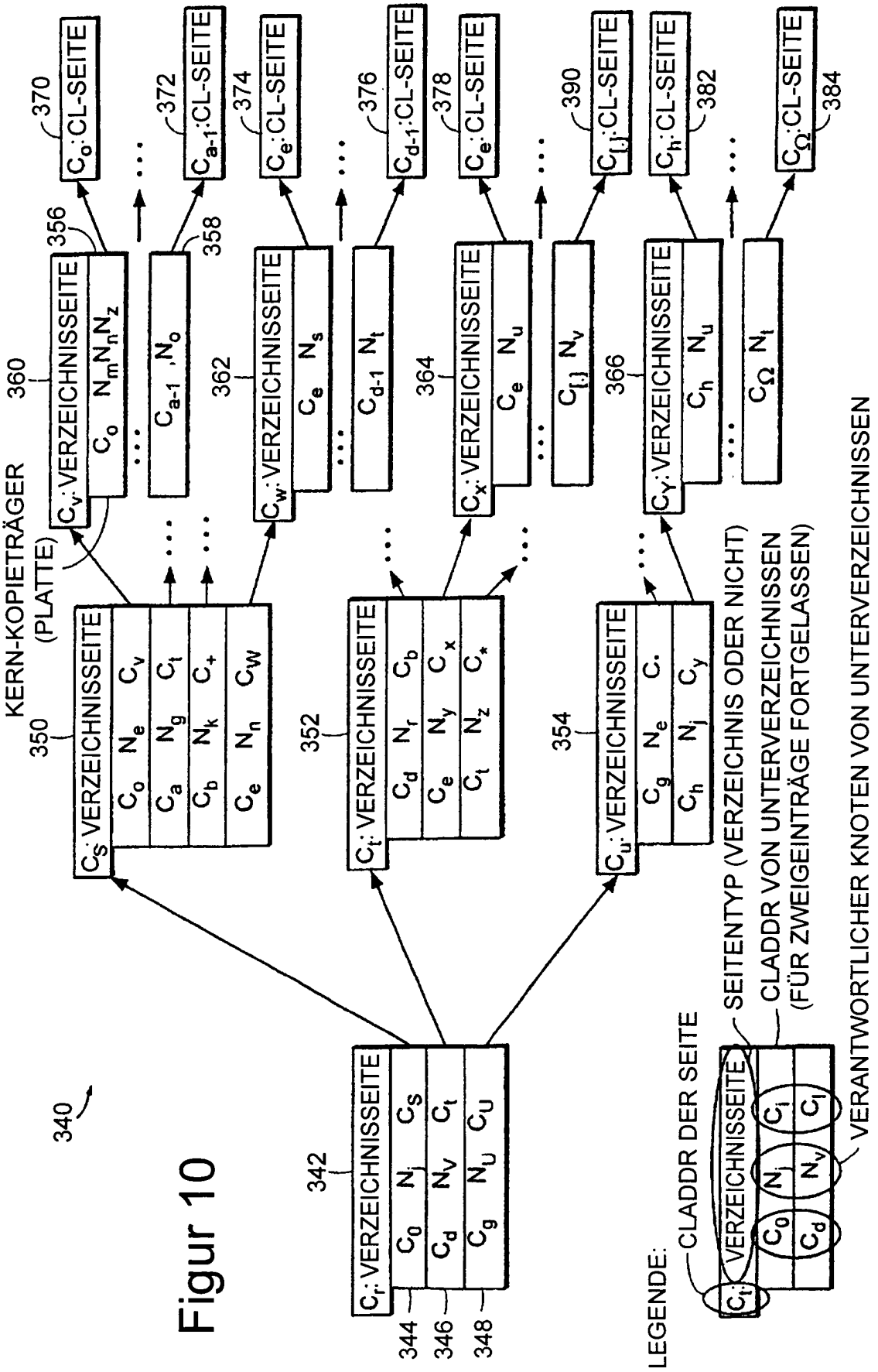
Figur 7


270
308
310
KONFIGURATIONS-DIENST
DIAGNOSTIK-DIENST
312
ANFORDERUNGEN
272
ABLAUFPLANUNGSEINRICHTUNG
CL-ELEMENT-AUFRUF UND RÜCKGABE
FERN-ANFORDERUNGEN
274
REKURSIVER ABLAUFAUFRUF
FERNOPERATIONEN
NETZWERK
304
276
LOKALER RAM-CACHE-SPEICHER
278
RAM-KOPIERSATZ
300
PHYSIKALISCHER SPEICHER
280
GLOBALES RAM-VERZEICHNIS
302
282
PLATTEN-KOPIERSATZ
GEMEINSAM VERWENDETE DATEN
284
GLOBALES PLATTENVERZEICHNIS
302
288
KONFIGURATIONS-VERWALTUNGSEINHEIT
290
VERFAHRENSSTEUEREINHEIT
GEMEINSAM VERWENDETE DATEN
294
LOKALER PLATTEN-CACHE-SPEICHER
298
PHYSIKA-LISCHES DATEI-SYSTEM

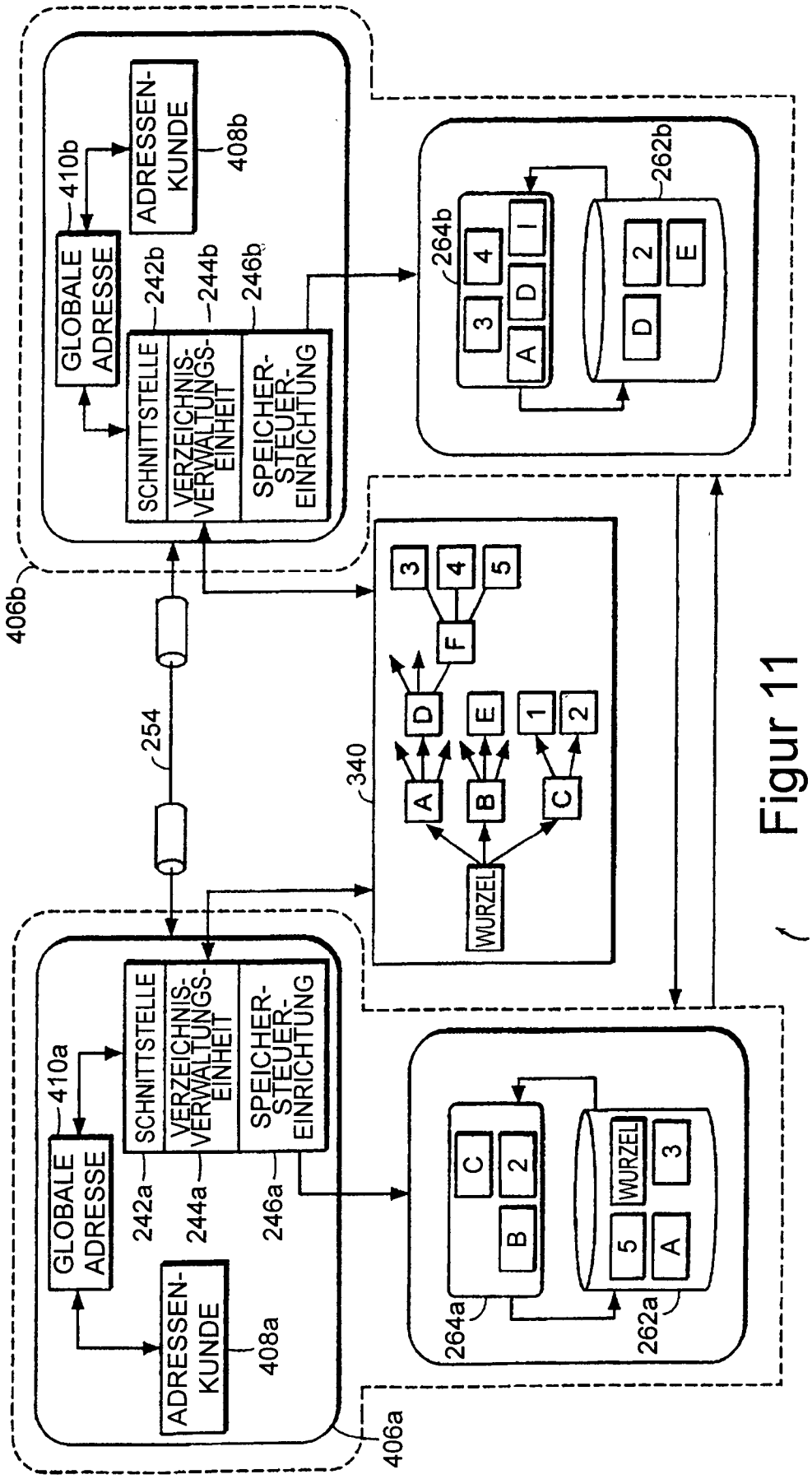
Figur 8


320
VERZEICHNISSEITE
322
SEITENKOPF
324
VERZEICHNISEINTRAG
ANFANG DES CL-ADRESSENBEREICHS
VERANTWORTLICHER KNOTEN DER ABHÄNGIGEN SEITE
CLADDR DER ABHÄNGIGEN SEITE
326
VERZEICHNISEINTRAG
ANFANG DES CL-ADRESSENBEREICHS
VERANTWORTLICHER KNOTEN DER ABHÄNGIGEN SEITE
CLADDR DER ABHÄNGIGEN SEITE
330
332
334
330
332
334
EINE VERZEICHNISSEITE BESTEHT AUS EINEM SEITENKOPF SOWIE EINEM ODER MEHREREN VERZEICHNISEINTRÄGEN
EIN VERZEICHNISEINTRAG WEIST DREI ATTRIBUTE AUF:
• EINE CLADDR, DIE DEN ANFANG DES SEITENBEREICHS BESCHREIBT, DEN DIESER EINTRAG ZU LOKALISIEREN HILFT
• DEN KNOTEN, DER MIT DEM VERFOLGEN DES EIGENTÜMERS DER ABHÄNGIGEN SEITE BEAUFTRAGT IST
• DIE CL-ADRESSE DER ABHÄNGIGEN SEITE (IN DEN ZWEIGEINTRÄGEN FORTGELASSEN)

Figur 9


KERN-KOPIETRÄGER
(PLATTE)
340
350
352
354
356
358
360
362
364
366
370
372
374
376
378
380
382
384
342
344
346
348
LEGENDE:
CLADDR DER SEITE
SEITENTYP (VERZEICHNIS ODER NICHT)
CLADDR VON UNTERVERZEICHNISSEN
(FÜR ZWEIGEINTRÄGE FORTGELASSEN)
VERANTWORTLICHER KNOTEN VON UNTERVERZEICHNISSEN

Figur 10


406b
GLOBALE ADRESSE
410b
ADRESSEN-KUNDE
408b
SCHNITTSTELLE
242b
VERZEICHNIS-VERWALTUNGS-EINHEIT
244b
SPEICHER-STEUER-EINRICHTUNG
246b
264b
262b
254
340
WURZEL
406a
GLOBALE ADRESSE
410a
ADRESSEN-KUNDE
408a
SCHNITTSTELLE
242a
VERZEICHNIS-VERWALTUNGS-EINHEIT
244a
SPEICHER-STEUER-EINRICHTUNG
246a
264a
262a
400

Figur 11