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

(10) **DE 697 21 641 T2** 2004.02.19

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 817 064 B1**
(21) Deutsches Aktenzeichen: **697 21 641.1**
(96) Europäisches Aktenzeichen: **97 304 515.6**
(96) Europäischer Anmeldetag: **25.06.1997**
(97) Erstveröffentlichung durch das EPA: **07.01.1998**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **07.05.2003**
(47) Veröffentlichungstag im Patentblatt: **19.02.2004**

(51) Int Cl.[7]: **G06F 12/08**

(30) Unionspriorität:
671303 01.07.1996 US

(84) Benannte Vertragsstaaten:
DE, FR, GB, IT, NL, SE

(73) Patentinhaber:
Sun Microsystems, Inc., Santa Clara, Calif., US

(72) Erfinder:
Hagersten, Erik E., Palo Alto, US; Hill, Mark Donald, Madison, US

(74) Vertreter:
Dr. Weber, Dipl.-Phys. Seiffert, Dr. Lieke, 65183 Wiesbaden

(54) Bezeichnung: **Verfahren und Vorrichtung für ein verzeichnisloses Speicherzugriffsprotokoll in einem Rechnersystem mit verteiltem gemeinsamen Speicher**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

[0001] Die vorliegende Erfindung bezieht sich auf Verfahren und eine Vorrichtung zur Erleichterung bzw. Unterstützung effektiver Kommunikation in einem Computernetzwerk.

[0002] Computernetzwerke mit verteilten, gemeinsam genutzten Speichern (Distributed Shared Memories, DSM) sind nach dem Stand der Technik bekannt. Zum Zwecke der Diskussion veranschaulicht **Fig.** 1 ein Computernetzwerk **10** mit einer Netzinfrastruktur **12** (NI). Vier Netzknoten **100**, **102**, **104** und **106** sind mit der Netzinfrastruktur **12** verbunden abgebildet. Durch die Netzinfrastruktur **12**, können die Knoten **100-106** untereinander kommunizieren, um Programme, Daten und ähnliches gemeinsam zu nutzen. Natürlich kann die Anzahl von Knoten, die pro Netzwerk vorgesehen sind, nach Bedarf variieren und jede beliebige Anzahl von Knoten umfassen.

[0003] Innerhalb jedes Netzknotens gibt es ein Speichermodul, auf dessen Speicherblöcke von anderen Netzknoten zugegriffen werden kann. In allgemeinen hat jeder Speicherblock im Netzwerk eine eindeutige Adresse, die es ermöglicht, ihn eindeutig zu adressieren. Die Vereinigung aller Speicherblöcke in den Knoten von Netzwerk **10** umfaßt den verteilten, gemeinsam genutzten Speicher (DSM). Es sollte jedoch beachtet werden, daß ein gegebener Speicherblock typischerweise einem Heimatknoten im Netzwerk **10** angegliedert ist, obwohl auf die Speicherblöcke des DSM von jedem beliebigen Netzknoten zugegriffen werden kann.

[0004] Für die Zwecke der vorliegenden Erfindung kann die Netzinfrastruktur **12** irgendeine Konfiguration haben und durch irgendein Protokoll implementiert werden. Allgemein besitzt die Netzinfrastruktur **12** die Fähigkeit, eine Nachricht von einem Knoten an einen anderen Knoten entsprechend der Zieladresse, die dieser Nachricht zugeordnet ist, korrekt zu übergeben. Ein Beispiel einer Netzinfrastruktur ist Sequent Numa-Q, erhältlich von Sequent Computer Systems, Inc. in Beaverton, OR (USA)

[0005] Jeder der Netzknoten **100-106** kann so einfach wie ein Computer mit einem einzelnen Prozessor sein, der an sein eigenes Speichermodul über einen Speichercache angeschlossen ist, Ein Netzknoten kann auch so kompliziert wie ein komplettes busbasiertes Mehrprozessorsystem oder sogar ein Mehrprozessornetzwerk sein. Im letzteren Fall kann ein Knoten mehrere Prozessoren enthalten, von denen jeder sowohl an sein eigenes Speichermodul und seinen eigenen Speichercache angeschlossen ist, als auch an den Speicher, der über andere Knoten im Netzwerk verteilt ist. Zur Vereinfachung der Darstellung wird die Erfindung hier mit Bezug auf einen Knoten mit einem einzelnen Prozessor beschrieben. Für die Fachleute auf dem Gebiet sollte es bei dieser Offenbarung offensichtlich sein, daß die Prinzipien und Techniken, die hier offenbart werden, leicht auf Knoten mit mehreren Prozessoren erweitert bzw. ausgedehnt werden können.

[0006] Nach dem Stand der Technik kommunizieren die Netzknoten typischerweise untereinander, indem sie einen busbasierten Ansatz oder ein direktes Protokoll verwenden. Als Beispiel zeigt **Fig.** 2 eine schematische Darstellung eines Computernetzwerks einschließlich der Beispielknoten **100a** und **100b** zur Implementierung einer Version eines busbasierten Protokolls nach dem Stand der Technik. Im Knoten **100a** von **Fig.** 2 ist der Prozessor **200a** mit einem Speichermodul **204a** verbunden, z. B. einem dynamischen Speichermodul mit wahlfreiem Zugriff (Dynamic Random Access Memory Module) über einen Speichercache **202a**, der typischerweise unter, Verwendung einer Art von schnellem Speicher, z. B. statischer Speicher mit wahlfreiem Zugriff (Static Random Access, SRAM). Das Speichermodul **204a** kann in Speicherblöcke unterteilt werden und der Speichercache **202a** dient dazu, den Zugriff auf die Speicherblöcke des Speichermoduls **204a** dadurch , zu beschleunigen, daß es eine Kopie des angeforderten Speicherblocks entweder von seinem eigenen Knoten oder von einem anderen Knoten im Netzwerk (wie zum Beispiel Knoten **100b**) in seinen schnellen Speicherschaltkreisen hält. Durch eine Netzwerkschnittstelle (in jedem Knoten enthalten, jedoch zur Vereinfachung der Darstellung nicht abgebildet) kann Knoten **100a** sowohl mit Knoten **100b** als auch mit anderen Knoten im Netzwerk über eine busbasierte Netzinfrastruktur, z. B. Bus **206**, kommunizieren, um Zugriff auf den verteilten, gemeinsam genutzten Speicher (DSM), der in den Knoten des Netzwerks verteilt ist, zu erlangen,

[0007] In einem busbasierten Computernetzwerk wird eine Speicheranforderung durch einen gegebenen Knoten typischerweise auf dem gemeinsamen Bus durch Rundsenden an anderen Knoten übermittelt, so daß die Anforderung von allen anderen Knoten im Netzwerk gesehen werden kann. Wenn zum Beispiel Prozessor **200a** von **Fig.** 2 Zugriff auf einen Speicherblock benötigt, der sich in einem anderen Speichermodul eines anderen Netzknotens befindet, übermittelt er typischerweise durch Rundsenden auf dem gemeinsamen Bus seine Speicherzugriffsanforderung. Alle Knoten in einem Netzwerk würden dieselbe Anforderung empfangen, und der Knoten, dessen Speicheradreßbereich zu der in der Speicherzugriffsanforderung mitgelieferten Speicheradresse paßt, antwortet dann.

[0008] Diese Rundsendetechnik funktioniert in adäquater Weise für relativ kleine Computernetzwerke. Mit zunehmender Größe und/oder physikalischer Ausbreitung von Computernetzwerken hat der busbasierte Ansatz einige Schwierigkeiten. Wenn zum Beispiel der Bus größer bzw. länger wird, um mehr Knoten aufzunehmen, ist es wünschenswert, den Bus mit einer höheren Geschwindigkeit zur betreiben, da jeder Knoten im allgemeinen auf den Bus für eine gewisse Zeitspanne zugreifen muß, um seine Transaktionen auszuführen. Einen größeren Bus mit

einer hohen Frequenz zur betreiben, ist schwierig, weil Busse mit dem Größenwerden elektrisch länger werden und elektrische Belange, z. B. Kapazität, ihre Betriebsfrequenz wesentlich beschränken können. Als Konsequenz daraus ist der busbasierte Ansatz im allgemeinen nicht für große oder physikalisch ausgedehnte Computernetzwerke geeignet.

[0009] Darüber hinaus erfordert der busbasierte Ansatz das Vorsehen eines Vermittlerschaltkreises, d. h. eines Mechanismus, der eine natürliche Reihenfolge der Transaktionen durch die verschiedenen Knoten des Computernetzwerkes erzwingt. Der Vermittlerschaltkreis muß sicherstellen, daß Speicherzugriffsanforderungen von verschiedenen Knoten richtig geordnet sind, um Konkurrenzsituationen mit nebenläufigen Zugriffen zu vermeiden. Die Verwendung eines Schiedsschaltkreises und eines Arbitrationsschemas bzw. Schemas zum Vermittlungsverfahren stellt eine zusätzliche Schicht bzw. Ebene von Komplexität dar, wodurch die Kosten bzw. der Aufwand bei der Implementierung und Instandhaltung bzw. Wartung des Computernetzwerkes erhöht werden.

[0010] Darüber hinaus stellt die große Anzahl von parallelen Nachrichten, die in einem busbasierten System vom anfordernden Knoten an alle Knoten in einem Netzwerk gesendet werden muß, eine zusätzliche Belastung der Busbandbreite dar. Das liegt daran, daß der anfordernde Knoten, wie zuvor erwähnt, jeden Knoten im Netzwerk abfragen und von jedem Knoten fordern muß, die Anforderung zu analysieren; um die Anforderung entweder zu ignorieren oder zu beantworten. Die zusätzliche Arbeit, die von den anderen Knoten im Netzwerk gefordert wird, stellt eine zusätzliche Verzögerung und zusätzliche Verarbeitung dar, die die Netzknoten ausführen müssen.

[0011] Die Verzeichnistechnik stellt einen Versuch dar, ein Computernetzwerk zu implementieren, in dem naturgemäßes Rundsenden nicht erforderlich ist, d. h. eine Transaktion oder eine Anforderung von einem Knoten braucht nicht in einer parallelen Weise auf einem gemeinsamen Bus an alle anderen Knoten im Netzwerk per Rundsenden verteilt zu werden. **Fig.** 3A veranschaulicht zum Zwecke der Diskussion einen Computernetzknoten **100** zur Implementierung des Verzeichnisprotokolls. Gemäß **Fig.** 3A ist darin ein Verzeichnis **210** abgebildet, das als eine Datenstruktur im Speicher implementiert werden kann und Verzeichniseinträge enthält, von denen jeder einem eindeutigen Speicherblock des Speichermoduls in Knoten **100** entspricht. Zum Beispiel ist im Verzeichnis **210** ein Verzeichniseintrag **212** abgebildet, der einem Speicherblock **208** in einem Speichermodul **204** entspricht. In jedem Knoten ist typischerweise ein Verzeichnis vorgesehen, das Verzeichniseinträge für die Speicherblöcke seines Speichermoduls enthält. Die Vereinigung aller Verzeichniseinträge in einem gegebenen Knoten stellt das Verzeichnis für diesen Knoten dar. In **Fig.** 3A ist auch eine Netzwerkschnittstelle **206** abgebildet, die den Schaltkreis zum Verbinden eines Knotens mit seiner Außenwelt, z. B. mit der Netzinfrastruktur, verkörpert.

[0012] Im Verzeichnisprotokoll muß jeder Knoten im Netzwerk, z. B. jeder der Knoten **100-106**, wissen, ob er eine exklusive Kopie eines Blocks des Speichers (eine modifizierbare oder M-Kopie), eine gemeinsam genutzte, nur lesbare Kopie (eine S-Kopie) hat oder ob er keine Kopie dieses Speicherblocks (eine ungültige oder I-Kopie) hat. Wenn ein Knoten eine M-Kopie des Blocks hat, sagt man, daß er über eine exklusive Kopie verfügt und diese Kopie modifizieren kann, um zu veranlassen, daß sie möglicherweise verschieden von ihrem Gegenpart im Speichermodul ihres Heimatknotens ist. Wenn irgendein Knoten im Computernetzwerk eine M-Kopie zum Beispiel von Speicherblock **208** besitzt, geben alle anderen Knoten ihre Kopien auf, d. h. sie besitzen nur I-Kopien von Speicherblock **208**.

[0013] Während nur ein Knoten eine M-Kopie eines Speicherblocks haben kann, können mehrere Knoten gemeinsam genutzte Kopien (S-Kopien) besitzen. Ein Knoten mit einer S-Kopie hat im Wesentlichen eine nur lesbare Kopie, d. h. er kann den Inhalt des Speicherblocks nicht verändern. S-Kopien eines Speicherblocks können gleichzeitig neben I-Kopien desselben Speicherblocks in einem Netzwerk vorliegen. S-Kopien eines Speicherblocks können jedoch nicht gleichzeitig mit irgendeiner M-Kopie desselben Speicherblocks existieren. Im allgemeinen wird von einem Knoten gesagt, daß er eine gültige Kopie eines Speicherblock hat, wenn er entweder eine S-Kopie oder eine M-Kopie dieses Speicherblocks hat.

[0014] Nach einer Ausführungsform kann ein Knoten anfordern, eine exklusive Kopie (M-Kopie) im Cache zu speichern, indem er eine RTO Anforderung aussendet, wobei RTO "read-to-own" ("lies um zu besitzen") bedeutet. Ein Knoten kann eine RTS-Anforderung aussenden, um anzufordern, daß eine gemeinsam genutzte Kopie eines Speicherblocks im Cache gespeichert wird, wobei RTS "readto-share" ("lies-um-gemeinsam-zu-nutzen") bedeutet. Ein Knoten kann auch anfordern, daß die exklusive M-Kopie eines Speicherblocks zurückgeschrieben wird, indem er eine WB Anforderung aus- sendet, wobei WB für write-back (schreib zurück) steht.

[0015] Wie gesagt weiß jeder Knoten in dem Computernetzwerk, welche Art von Kopie eines Speicherblocks **208** er aktuell besitzt. Daher wird dieses Wissen bezüglich des Speicherblocks **208** zwischen den Knoten im Netzwerk verteilt. In Übereinstimmung mit dem Verzeichnisprotokoll ist dasselbe Wissen bezüglich des Speicherblocks **208** auch auf dem Heimatknoten des Speicherblocks **208** zentralisiert, d. h. in einem Verzeichniseintrag **212** von Verzeichnis **210**.

[0016] Um die Darstellung zu vereinfachen, wird die Diskussion hier unter Bezug auf das Vier- Knoten-Netzwerk von **Fig.** 1 geführt, obwohl, wie zuvor bemerkt, ein Computernetzwerk jede beliebige Anzahl von Knoten enthalten kann. Für ein Vier- Knoten-Netzwerk kann ein Verzeichniseintrag **212** wie in **Fig.** 3B abgebildet die Verzeichniszustände **220-226**

enthalten, welche die Kopien eines Speicherblocks **208** repräsentieren, die in den entsprechenden Knoten **100-106** vorliegen. Gemäß dem Verzeichniseintrag von **Fig.** 3B hat Knoten **100** aktuell eine exklusive M-Kopie des Speicherblocks **208** (dargestellt durch den M-Zustand **220**) und alle anderen Knoten **102**, **104** und **106** des Computernetzwerks haben ungültige I-Kopien des Speicherblocks **208** (dargestellt durch die I-Zustände **222**, **224** und **226**).

[0017] Gemäß dem Verzeichniseintrag von **Fig.** 3C hat Knoten **104** nun die exklusive M-Kopie des Speicherblocks **208** (dargestellt durch den M-Zustand **234**) und alle anderen Knoten **100**, **102** und 106 des Computernetzwerks haben I-Kopien des Speicherblocks **208** (dargestellt durch die I-Zustände **230**, **232** und **236**). In dem Verzeichniseintrag von **Fig.** 3D haben die Knoten **100**, **102** und **104** gemeinsam genutzte S-Kopien des Speicherblocks **208** (dargestellt durch die S-Zustände **240**, **242** und **244**), während Knoten **106** keine Kopie des Speicherblocks **208** hat (dargestellt durch den I-Zustand **246**).

[0018] Darüber hinaus ist ein Anhängig- bzw. Pending-Flag **213** bei jedem Verzeichniseintrag **212** vorgesehen. Das Pending-Flag wird gesetzt, wann immer eine Transaktion anhängig ist, die sich auf einen bestimmten Speicherblock bezieht. Das Pending-Flag **213** bleibt gesetzt, bis die Transaktion abgeschlossen ist; zu diesem Zeitpunkt wird es zurückgesetzt, um zu erlauben, daß eine nachfolgende Transaktion, die sich auf denselben Speicherblock bezieht, bedient wird.

[0019] In Übereinstimmung mit dem Verzeichnisprotokoll nach dem Stand der Technik wird die Speicherzugriffsanforderung von der Netzinfrastruktur (NI) 12 an dem Heimatknoten, d. h. an den Knoten, der den Speicheradreßraum enthält, in den die Adresse des angeforderten Speicherblocks abgebildet wird, weitergeleitet, wenn irgendein Knoten des Computernetzwerks **10** eine exklusive oder gemeinsam genutzte Kopie eines Speicherblocks anfordert. Zum Zweck der Diskussion nehmen wir an, daß der Speicherblock **208** von Knoten **100** von einem anderen Netzknoten angefordert wurde. Sobald der Heimatknoten, d. h. Knoten **100**, die Anforderung empfängt, befragt er den Verzeichniseintrag **212**, der dem Speicherblock **208** zugeordnet ist, um den aktuellen Zustand des Speicherblocks **208** in den verschiedenen Knoten des Netzwerks sicherzustellen.

[0020] Wenn der aktuelle Zustand des Speicherblocks **208** zum Beispiel so ist, wie in **Fig.** 3B dargestellt, und der Knoten **104** anschließend eine exklusive M-Kopie des Speicherblocks **208** anfordert, wird die Anforderung von der Netzinfrastruktur 12 an die Netzwerkschnittstelle **206** des Heimatknotens **100** gesandt (siehe **Fig.** 3A). Wenn der Knoten **100** die RTO Anforderung von Knoten **104** empfängt, befragt er sein Verzeichnis **210** und stellt anhand des Verzeichniseintrags **212** (dessen Zustände in **Fig.** 3B abgebildet sind) fest, daß der Heimatknoten **100** aktuell die exklusive M-Kopie des Speicherblocks **208** besitzt. Da der Heimatknoten **100** bereits die einzige gültige Kopie des Speicherblocks **208** im Netzwerk hat, kann der Heimatknoten **100** sofort eine Kopie des Speicherblocks **208** an den anfordernden Knoten **104** senden und seinen Verzeichniseintrag **212** aktualisieren, um dem zu entsprechen, was in **Fig.** 3C dargestellt ist, d. h. die, Tatsache widerzuspiegeln, daß der Knoten **104** nun die exklusive M-Kopie des Speicherblocks **208** besitzt, die Kopie im Knoten **100** auf eine I-Kopie zurückgestuft wurde und Knoten **102** und **106** weiterhin I-Kopien haben. Sobald der anfordernde Knoten **104** seine M-Kopie erhält, sendet er eine Fertigstellungs- bzw. Abschlußnachricht an den Heimatknoten **100**, um das Pending-Feld des Verzeichniseintrags **212** zurückzusetzen, um zu ermöglichen, daß darauf folgende Transaktionen, die den Speicherblock **208** betreffen, bedient werden.

[0021] Als ein weiteres Beispiel wird im Fall, daß der Knoten **102** anschließend eine RTS-Transaktion für Speicherblock **208** aussendet, um eine gemeinsam genutzte S-Kopie anzufordern, die RTS-Anforderung von Knoten **102** von der Netzinfrastruktur **12** an den Heimatknoten von Speicherblock **208**, d. h. den Knoten **100**, weitergeleitet. Angenommen daß der aktuelle Zustand des Speicherblocks **208** so ist, wie in **Fig.** 3C abgebildet, dann kann der Heimatknoten **100** anhand des Verzeichniseintrages **212** herausfinden, daß der Knoten **104** aktuell die einzige exklusive Kopie von Speicherblock **208** hat. Er sendet daraufhin eine Anforderung an den Knoten **104**, um Knoten **104** aufzufordern, eine Kopie von Speicherblock **208** an den anfordernden Knoten **102** zu senden. Der Heimatknoten **100** kann auch anfordern, daß der Knoten **104** seine Kopie von einer I-Kopie zu einer S-Kopie aktualisiert. Zum selben Zeitpunkt kann der Heimatknoten **100** seinen eigenen Verzeichniseintrag **212** aktualisieren, um den neuen Zustand von Speicherblock **208** in bzw. bei Knoten **104** widerzuspiegeln. Sobald der Knoten **102** eine Kopie von Speicherblock **208** empfängt, wird sein Zustand im Verzeichniseintrag **212** von einer I-Kopie zu einer S-Kopie (Zustand **242**) aktualisiert.

[0022] Bei einigen Implementierungen, zum Beispiel der Speicherspiegelungstechnik, hat der Heimatknoten, z. B. Knoten **100** in diesem Beispiel, wann immer es eine S-Kopie in irgendeinem Knoten des Netzwerks gibt, auch eine gemeinsam genutzte Kopie von diesem Speicherblock (S-Kopie). Auf diese Art und Weise kann der Heimatknoten schnell die nächste Anforderung einer gemeinsam genutzten Kopie bedienen, ohne einen anderen Knoten im Netzwerk auffordern zu müssen, eine gemeinsam genutzte Kopie an den anschließend anfordernden Knoten zu übermitteln. In Übereinstimmung mit einer solchen Implementienng erhält der Heimatknoten **100** auch eine S-Kopie des Speicherblocks **208** und der Zustand **240** wird vom I-Zustand (Zustand **230**) von **Fig.** 3C zu einem S-Zustand in **Fig.** 3D höhergestuft.

[0023] Man erkennt, daß hier nur drei Zustände

(MIS/T) und drei Arten von Transaktionen (RTO/RTS/WB) diskutiert werden, um die Darstellung zu vereinfachen. Natürlich kann es andere bzw. weitere Zustände; Transaktionen und Variationen bzw. Änderungen bei der Implementierung geben. Es sollte auch beachtet werden, daß das Vorhandensein des Verzeichnisses die Notwendigkeit beseitigt bzw. eliminiert, eine Speicherzugriffsanforderung durch Rundsenden von einem Knoten an alle Knoten des Netzwerks zu senden, da der Heimatknoten immer seine Verzeichniseinträge befragen kann, um den genauen Knoten, von dem eine Kopie erhalten werden kann, festzustellen und direkt diesen Knoten auffordern kann, eine Kopie an den anfordernden Knoten zu schicken. Wenn nötig, kann der Heimatknoten direkt einen anderen Knoten im Netzwerk auffordern, seine Kopie des angeforderten Speicherblocks zu ändern, um den Anforderungen des Protokolls zu genügen, z. B. auf eine I-Kopie zurückzustufen, wenn es eine M-Kopie anderswo im Netzwerk gibt.

[0024] Die Verwendung des Pending-Flags, z. B. Bit **213** von **Fig.** 3A, beseitigt die Notwendigkeit für irgendeine natürliche Reihenfolge in dem Netzwerk. Mit anderen Worten stellt die Verwendung; des Pending-Flags sicher, daß die aktuelle Transaktion für einen gegebenen Speicherblock abgeschlossen ist, bevor die nächste Transaktion, die diesen Speicherblock betrifft, bedient wird. Wenn mehrere Transaktionen bezüglich desselben Speicherblocks vom Heimatknoten empfangen werden, können sie zum Beispiel in der Reihenfolge ihres Empfangs innerhalb des Netzwerkschnittstellenschaltkreises **206** in eine Warteschlange gestellt werden, um der Reihe nach bedient zu werden.

[0025] Obwohl das Verzeichnisprotokoll die Notwendigkeit einer natürlichen Reihenfolge und des selbstverständlichen bzw. naturgegebenen Rundsendens in einem Computernetzwerk beim Bedienen bzw. Bearbeiten von Speicherzugriffsanforderungen beseitigt, stellt die Notwendigkeit eines Verzeichniseintrages für jeden Speicherblock in einem Knoten einen beträchtlichen Speicheroverhead bzw. -zusatzaufwand dar. Dieser Speicheroverhead kann für Knoten mit einer großen Anzahl von Speicherblöcken ganz beträchtlich werden. Darüber hinaus erfordert das Verzeichnisprotokoll zusätzlichen Aufwand auf der Seite des Heimatknotens, um die Zustände seiner Speicherblöcke im gesamten Computernetzwerk nachzuhalten bzw. zu verfolgen. Dieses Erfordernis stellt eine zusätzliche Komplexitätsebene bei der Implementierung und der Verwaltung des Computernetzwerkes dar.

[0026] Ein eingeschränktes Verzeichnisschema wurde von Kumar et al. in dem Artikel bzw. Beitrag mit dem Titel "Efficient and Scalable Cache Coherence Schernes for Shared Memory Hypercache Multiprocessors" aus Proc. of Supercomputing 94, IEEE Comput. Soc. Press., 1994, Seiten 498-507 vorgeschlagen. Das Schema ist eine Mischform zwischen den verzeichnisartigen und pufferartigen Schemata. In dem ZPRB-Schema besteht der Verzeichniseintrag im Heimatknoten genau aus dem Zustand eines Speicherblocks. Es werden keine Zeiger oder -Vektoren auf dem Laufenden gehalten, um die Identität der Knoten zusammen mit einer Kopie des Blockes zu halten bzw. zu speichern. Das Schema hat einen reduzierten Speicheroverhead verglichen mit vollständigen Verzeichnisschemata. Jedoch erfordert es immer noch mehr Speicheroverhead als ein verzeichnisloses Schema. Um sowohl Skalierbarkeit als auch effiziente Kohärenzmechanismen zu erreichen, implementiert das ZPRB-Schema das Mitlesen bzw. Snooping auf Punkt-zu-Punkt-Netzwerken. Dies erfolgt durch Verwenden eines eingebetteten Rings, um einige der Steuernachrichten zu versenden, während die Datennachrichten weiterhin das Routing nach kürzestem Pfad verwenden. Die Ungültigkeitsnachrichten für einen gemeinsam genutzten Block werden über den Ring versendet, der in das Netzwerk eingebettet bzw. eingeschlossen ist und der von allen Knoten an dem Ring mit gelesen bzw. abgetastet wird. Unreine bzw. geänderte Blöcke (dirty blocks) werden durch Aussenden von Suchnachrichten über den Ring lokalisiert. Die verwendete Verbindung untereinander ist ein Hyperkubus-Netzwerk und ein Ring ist in dieses Netzwerk eingefügt, indem ein binäres Codierschema verwendet wird, das gespiegelter Gray-Code bzw. Reflected Gray Code genannt wird. Um das Mitlesen bzw. Snooping auf dem eingebetteten Ring zu unterstützen, wird daher bei jedem Verbindungseingang ein zusätzlicher Nachrichtenpuffer in die Kommunikationssteuerung aufgenommen. Eine ankommende Nachricht wird zusätzlich dazu, daß sie an die Ausgangsverbindung bzw. aus- bzw. abgehende Verbindung weitergeleitet wird, auch in diesen Puffer kopiert. Sobald die vollständige Nachricht kopiert ist, wird sie an die Cachesteuerung weitergeleitet, die eine geeignete Aktion vornimmt. Diese Aktion könnte das Ungültigmachen eines Blocks oder eine Antwort auf eine Leseanforderung, etc. sein. Dies ist logisch ähnlich zum Mitlesen in einem busbasierten System, da jeder Knoten die Nachricht zu sehen bekommt. Pending-Zustände, die beim Heimatknoten an Speicher-Blöcken auf eine Blockierungsanforderung hin gesetzt und beim Abschluß dieser Anforderung freigegeben werden, halten die Kohärenz für Anforderungen aufrecht, die außer der Reihe auftreten.

[0027] Mit Blick auf das Voranstehende sind Verfahren und eine Vorrichtung wünschenswert, die es Knoten eines Computernetzwerkes erlauben, auf den verteilten, gemeinsam genutzten Speicher des Netzwerkes in einer effizienten Art und Weise zuzugreifen.

[0028] Verschiedene diesbezügliche Aspekte und Eigenschaften der Erfindung sind in den angefügten Patentansprüchen definiert.

[0029] Ausführungsformen der Erfindung beziehen sich auf verbesserte Techniken, die es Knoten eines Computernetzwerkes erlauben, auf den verteilten,

gemeinsam genutzten Speicher (Distributed Shared Memory, DSM) des Netzwerkes in einer effizienten Art und Weise zuzugreifen.

[0030] Die Erfindung bezieht sich nach einer Ausführungsform auf ein Verfahren wie in Anspruch 1 dargelegt.

[0031] Nach einer weiteren Ausführungsform bezieht sich die Erfindung auf eine Vorrichtung wie in Anspruch 7 dargelegt.

[0032] Die Erfindung wird jetzt anhand von Beispielen unter Bezug auf die beigefügten Zeichnungen beschrieben, in denen gleiche Teile mit den gleichen Bezugszeichen bezeichnet werden und in denen:

[0033] **Fig.** 1 ein Computernetzwerk mit einer Mehrzahl von Knoten und einer Netzinfrastruktur zum Zweck der Diskussion veranschaulicht.

[0034] **Fig.** 2 eine schematische Darstellung eines Computernetzknotens zur Implementierung einer Version des busbasierten Protokolls nach dem Stand der Technik ist.

[0035] **Fig.** 3A zum Zweck der Diskussion einen Computernetzknoten zur Implementierung des Verzeichnisprotokolls nach dem Stand der Technik veranschaulicht.

[0036] **Fig.** 3B-3D verschiedene Verzeichniseinträge zeigen, die Verzeichniszuständen eines gegebenen Speicherblocks in den Knoten des Computernetzwerks von **Fig.** 1 zu verschiedenen Zeitpunkten entsprechen.

[0037] **Fig.** 4 in Übereinstimmung mit einem Aspekt der vorliegenden Erfindung eine schematische Darstellung eines Computernetzknotens zeigt, der zur Implementierung des verzeichnislosen Protokolls geeignet ist.

[0038] **Fig.** 5A und 5B in Übereinstimmung mit verschiedenen Aspekten der vorliegenden Erfindung die Organisation einer Pending-Flag-Struktur veranschaulichen.

[0039] **Fig.** 6 zum Zweck der Diskussion und in Reaktion auf verschiedene Speicherzugriffsanforderungen die Zustände eines Speicherblocks in den Knoten des Netzwerkes von **Fig.** 1 veranschaulicht.

[0040] **Fig.** 7 ein Zustandsdiagramm enthält, das die verschiedenen Kommunikationspfade zwischen einem anfordernden Knoten, einem Heimatknoten und einem abhängigen Knoten beim Bedienen von Speicherzugriffsanforderungen veranschaulicht.

[0041] Ausführungsformen einer Erfindung, die sich auf ein verzeichnisloses Protokoll beziehen, werden beschrieben, um es Knoten eines Computernetzwerks zu erlauben, auf den verteilten, gemeinsam genutzten Speicher des Netzwerks in einer effizienten Art und Weise zuzugreifen. In der folgenden Beschreibung sind zahlreiche spezifische Details dargelegt, um für ein umfassendes Verständnis der vorliegenden Erfindung Vorkehrung zu treffen. Es ist jedoch für die Fachleute auf diesem Gebiet offensichtlich, daß die vorliegende Erfindung ohne einige oder alle diese spezifischen Details praktiziert bzw: angewandt werden kann. Bei anderen Gelegenheiten bzw. Beispielen wurden wohlbekannte Strukturen und Prozeßschritte nicht im Detail beschrieben, um die vorliegende Erfindung nicht unnötig zu verschleiern.

[0042] Gemäß einer Ausführungsform der vorliegenden Erfindung werden eine verbesserte Vorrichtung und verbesserte Verfahren zum Bedienen von Speicherzugriffsanforderungen von Netzknoten auf Speicherblöcke des verteilten, gemeinsam genutzten Speichers des Netzwerks bereitgestellt. Die Speicherzugriffsanforderungen können bedient werden, ohne einen natürlichen bzw. vorgegebenen Reihenfolgenmechanismus wie zum Beispiel einen Vermittler und ohne ein naturgemäßes Rundsenden an alle Knoten in dem Computernetzwerk für jede Speicherzugriffsanforderung zu benötigen. Die Beseitigung des vorgegebenen Reihenfolgenmechanismus reduziert die Komplexität der Netzinfrastruktur. Die Beseitigung des Erfordernisses des naturgemäßen Rundsendens an alle Netzknoten für jede Speicherzugriffsanforderung reduziert vorteilhafterweise die Anzahl von Transaktionen, welche die Netzinfrastruktur behandeln muß und verbessert dadurch die Bandbreite des Netzwerkes. Im Gegensatz zum Verzeichnisprotokoll nach dem Stand der 'Technik erfordert die vorliegende verzeichnislose Erfindung keine Implementierung und Aufrechterhaltung von Knotenverzeichnissen in den Netzknoten. Die Beseitigung eines zentralisierten Verzeichnisses bei jedem Knoten reduziert vorteilhafterweise sowohl den Speicheroverhead als auch die Komplexität, die mit dem Verzeichnisprotokoll einhergeht.

[0043] **Fig.** 4 zeigt in Übereinstimmung mit einer Ausführungsform der vorliegenden Erfindung eine schematische Darstellung eines Computernetzwerkknotens wie zum Beispiel den Knoten **400** aus **Fig.** 1, der für die Implementierung des verzeichnislosen Protokolls der vorliegenden Erfindung geeignet ist. In **Fig.** 4 ist der Knoten **100** mit einem Prozessor **400** abgebildet. Der Prozessor **400** ist an einen Speichercache **402** angeschlossen, der verwendet wird, um die angeforderten Speicherblöcke im Cache zu speichern bzw. zwischenzuspeichern. Die von dem Speichercache **402** zwischengespeicherten Speicherblöcke stellen Speicherblöcke dar, die vom Knoten **100** angefordert wurden. Der Heimatknoten eines im Cache gespeicherten Speicherblocks kann derselbe sein wie der Knoten, auf dem der verzeichnislose Mechanismus gehalten wird, oder ein anderer Knoten in dem Netzwerk.

[0044] Sowohl der Speichercache **402** als auch das Speichermodul **404** sind an eine Netzwerkschnittstelle **406** angeschlossen. Die Netzwerkschnittstelle **406** stellt den Schaltkreis dar, der es dem Knoten **100** ermöglicht, mit der Netzinfrastruktur zu kommunizieren, wie zum Beispiel der Netzinfrastruktur **12** von **Fig.** 1. Das Speichermodul **404** hat typischerweise einen Adreßbereich, der verschieden ist von den Adreßbereichen, die anderen Speichermodulen in dem Netzwerk zugeordnet sind. Die Vereinigung aller

Speichermodule **404** in einem Netzwerk machen den verteilten, gemeinsam genutzten Speicher (DSM) aus. Das Speichermodul **404** enthält ferner alle Speicherblöcke eines Knotens. Wenn das Netzwerk ruhig bzw. untätig ist, hat jeder Speicherblock einen Heimatknoten. Wie zuvor erwähnt kann der Knoten **100** einfach wie ein einzelner Prozessor mit einem Cache oder o kompliziert wie ein komplettes busbasiertes Mehrprozessorsystem sein.

[0045] Um das verzeichnislose Protokoll zu implementieren wird gemäß einer Ausführungsform der vorliegenden Erfindung eine Pending-Flag-Struktur **410** bereitgestellt. Nach einer Ausführungsform ist die Pending-Flag-Struktur **410** an die Netzwerkschnittstelle **406** angeschlossen. Die Pending-Flag-Struktur **410** stellt die Speicherstruktur zum Speichern des Flags dar, das anzeigt, ob auf einen bestimmten Speicherblock, z. B. Speicherblock **408**, von dem Netzknoten aktuell zugegriffen wird und er von einem anderen Knoten nicht gelesen oder beschrieben werden sollte. Wenn das Pending-Flag gesetzt ist; ist der Zugriff auf den zugeordneten Speicherblock für andere Netzknoten verboten. Wenn das Pending-Flag zurückgesetzt ist, ist der zugeordnete Speicherblock verfügbar, um die nächste Transaktion, die ihn anfordert, zu bedienen.

[0046] Nach einer Ausführungsform kann die Pending-Flag-Struktur **41A** durch das Bereitstellen eines Flags (welches ein Bit sein kann; obwohl auch mehrere Bits verwendet werden können) für jeden Speicherblock des Speichermoduls **404** implementiert werden. Eine vordefinierte Beziehung zwischen den Adressen der Speicherblöcke im Speichermodul **404** und den Adressen der Flags in der Pending-Flag-Struktur kann es Knoten **100** ermöglichen, den eindeutigen Speicherblock, der jedem Flag zugeordnet ist, zu ermitteln. Zur Veranschaulichung kann, wenn die Adresse im Speichermodul **404** um vier erhöht wird, die Adresse in der Pending-Flag-Struktur zum Beispiel um eins erhöht werden. **Fig.** 5A veranschaulicht diese Implementierung, wobei die Pending-Flags **420-426** unter Verwendung irgendeiner herkömmlichen Abbildungstechnik auf die Speicherblöcke des Speichermoduls **404** abgebildet werden.

[0047] Nach einer weiteren Ausführungsform der vorliegenden Erfindung ist es nicht erforderlich, daß jeder Speicherblock ein zugeordnetes Pending-Flag in der Pending-Flag-Struktur **410** hat. In Übereinstimmung mit diesem Aspekt, der in **Fig.** 5B dargestellt ist, ist jeder Pending-Struktur-Eintrag **430** in der Pending-Flag-Struktur **410** durch ein Gültig-Flag-Feld **432** und ein Adreßfeld **434** implementiert. Wenn ein Speicherblock "anhängig" bzw. "pending" ist, wird seine Adresse in das Adreßfeld **434** des Pending-Struktur-Eintrags **430** eingetragen und das Gültig-Flag im Gültig-Flag-Feld **432** gesetzt. Die Anzahl von Pending-Struktur-Einträgen **430** in der Pending-Flag-Struktur **410** kann kleiner sein als die Anzahl von Speicherblöcken im Speichermodul **404**, um den Speicheroverhead zu reduzieren. Die Implementierung aus **Fig.** 5B ist vorteilhaft für Netzknoten, die ein relativ großes Speichermodul **404** haben, da nur eine kleine Anzahl von Speicherblöcken typischerweise zu jedem beliebigen gegebenen Zeitpunkt im Cache gespeichert werden und die Anzahl von Pending-Struktur-Einträgen **430** relativ klein gehalten werden kann. Die Anzahl von Pending-Struktur-Einträgen **430** kann von System zu System in großem Maße variieren abhängig von der Anzahl von Speicherblöcken, von denen erwartet wird, daß sie zu irgendeinem beliebigen gegebenen Zeitpunkt im Cache gespeichert werden (wobei die Anzahl für ein bestimmtes Netzwerk statistisch bestimmt werden kann).

[0048] Der Betrieb der vorliegenden verzeichnislosen Technik kann am besten unter Bezug auf die Beispiele von **Fig.** 6 und das Zustandsdiagramm von **Fig.** 7 veranschaulicht werden. In Figur 6 sind in den Zeilen A-H die Zustände für den Speicherblock **408** von Knoten **100** aus **Fig.** 4 in den vier Knoten des Netzwerks dargestellt. Es sollte berücksichtigt werden, daß das Speichern im Cache typischerweise für eine Mehrzahl von Speicherblöcken durchgeführt wird, auch wenn hier ein einzelner Speicherblock **408** im Detail diskutiert wird, um die Darstellung zu vereinfachen. Es sollte auch berücksichtigt werden, daß die Anzahl von Knoten in einem Netzwerk gemäß dem Bedarf variiert werden kann. Darüber hinaus ist die Matrix von **Fig.** 6 hier nur zum Zweck der Diskussion dargestellt. In dem verzeichnislosen Protokoll ist es nicht nötig, eine zentralisierte Matrix im Computerspeicher zu erzeugen und auf dem Laufenden zu halten.

[0049] In Zeile A wird dargestellt, daß Knoten **100** eine exklusive M-Kopie von Speicherblock **408** hat (Zustand M' in Zeile A, Spalte 100). Aufgrund der Definition müssen alle andere Netzknoten ungültige Kopien von Speicherblock **408** haben (dargestellt durch Zustand 1 in Zeile A, Spalten 102-106). im Übrigen kann die M-Kopie von Speicherblock **408** aktuell vom Speichercache in seinem Heimatknoten **100** zwischengespeichert werden oder kann sich in dem Speichermodul des Heimatknotens befinden.

[0050] Transaktion No. 1 (Zeile A zu Zeile B von **Fig.** 6): Wir nehmen an, daß Knoten **104** nun eine exklusive M-Kopie von Speicherblock **408** wünscht, die sich, wie in Zeile A dargestellt, aktuell bei seinem Heimatknoten **100** befindet. Gemäß **Fig.** 7 stellt Knoten **104** den anfordernden Knoten **702** dar, während Knoten **100** den Heimatknoten für den Speicherblock **408**, darstellt, der in **Fig.** 7 als Heimatknoten **708** abgebildet ist. Der abhängige Knoten **712** stellt den Knoten dar, bei dem sich die Kopie von Speicherblock **408** aktuell befindet. In Zeile A ist der abhängige Knoten **712** zufällig derselbe Knoten wie der Heimatknoten, d. h. Knoten **100**.

[0051] Die RTO-Anforderung von Knoten **104** (dem anfordernden Knoten **702** in der ersten Transaktion) wird an den Heimatknoten **100** (Knoten **708**) über Pfad **704** weitergeleitet. Das Weiterleiten der

RTO-Transaktion vom anfordernden Knoten an den Heimatknoten wird typischerweise von der Netzinfrastruktur **12** unter Verwendung der Adresse, die mit der RTO- Anforderung geliefert wird, behandelt. Die Netzinfrastruktur **12** weiß, wo der Heimatknoten für einen bestimmten Speicherblock ist, zum Beispiel durch Abbildung seiner Adresse auf die Adreßbereiche der verschiedenen Knoten. Wenn der Heimatknoten **100** (Knoten **708**) die RTO-Nachricht empfängt, setzt er das Pending-Flag, das dem angeforderten Speicherblock **408** zugeordnet ist. Darüber hinaus weiß er, daß er eine exklusive M-Kopie von Speicherblock **408** besitzt und kann daher folgern, daß alle andere Knoten eine ungültige Kopie von Speicherblock **408** haben müssen. Da er auch der Knoten ist, an dem sich die Kopie befindet (abhängiger Knoten **712**), kann Knoten **100** in **Fig.** 7 betrachtet werden, als wenn er sowohl den Heimatknoten **708** als auch den abhängigen Knoten **712** umfaßt.

[0052] Der Knoten **100** (Heimatknoten **708**/abhängiger Knoten **712**) sendet danach eine Kopie des angeforderten Speicherblocks **408** über Pfad **714** an den anfordernden Knoten **104** (Knoten **702**). Beim Empfang einer Kopie von Speicherblock **408** aktualisiert der anfordernde Knoten **104** (Knoten **702**) seine Kopie zu einer exklusiven M-Kopie und sendet eine Bestätigungsnachricht über Pfad **706** an den Heimatknoten **100** (Knoten **708**). Der Empfang der Bestätigungsnachricht durch den Heimatknoten **100** (Knoten **708**) erlaubt es, daß das Pending-Flag, das dem Speicherblock **408** zugeordnet ist, zurückgesetzt wird, wodurch es möglich wird, daß nachfolgende Transaktionen, die den Speicherblock **408** einbeziehen, bedient werden können. Wie in Transaktion No. 1 gezeigt, beseitigt die Verwendung des Pending-Flags und expliziter bzw. ausdrücklicher Nachrichten zwischen dem anfordernden Knoten, dem Heimatknoten und dem abhängigen Knoten (über die Pfade **704**, **706**, **710** und **714**) die Notwendigkeit einer netzwerkweiten Rundsendung, um diese RTO-Transaktion No. 1 zu bedienen.

[0053] Darüber hinaus beseitigt die Verwendung des Pending-Flags die Notwendigkeit eines vorgegebenen bzw. natürlichen Reihenfolgemechanismus, da Transaktionen von dem empfangenden Heimatknoten in der Reihenfolge, in der sie empfangen werden, in eine Warteschlange gestellt und in dieser Reihenfolge bedient werden können, wann immer das Pending-Flag zurückgesetzt wird. Wie Fachleute auf diesem Gebiet verstehen, ist die Anzahl der im Netzwerk versendeten Nachrichten vorteilhafterweise für Transaktion No. 1 in dem verzeichnislosen Protokoll dieselbe wie in der Implementierung, die das Verzeichnisprotokoll einschließt, d. h. es ist keine Leistungseinbuße mit der Verwendung des verzeichnislosen Protokolls für Transaktion No. 1 verbunden. Vorteilhafterweise wird jedoch sowohl das Verzeichnis als auch die es begleitende Komplexität und der Speicher/Transaktionsoverhead in dem verzeichnislosen Protokoll der Erfindung nicht benötigt.

[0054] Transaktion No. 2 (Zeile B zu Zeile D): In der Transaktion No. 2 fungiert der Knoten **102** als der anfordernde Knoten und fordert eine exklusive Kopie von Speicherblock **408** an, indem er eine RTO-Transaktion aussendet. Die RTO-Transaktion wird von der Netzinfrastruktur **12** an den Heimatknoten **100** von Speicherblock **408**, d. h. Knoten **708** in Transaktion No. 2, über Pfad **704** weitergeleitet, und führt dazu, daß der Heimatknoten **100** das dem Speicherblock **408** zugeordnete Pending-Flag setzt.

[0055] Da der Knoten **100** nur weiß, daß er eine ungültige Kopie von Speicherblock **408** besitzt (Zeile B, Spalte **1A0**) und er in dem verzeichnislosen Protokoll keinen Verzeichniseintrag zum Befragen hat, weiß der Knoten **100** nicht, wo im Netzwerk eine gültige Kopie von Speicherblock **408** gefunden werden kann: Jedoch muß der Knoten **100** den Knoten, der aktuell die Kopie von Speicherblock **408** besitzt, auffordern, diese Kopie an den anfordernden Knoten **102** weiterzuleiten. Folglich sendet der Heimatknoten **100** (Knoten **708**) eine Anforderung über Pfad **710**, nach einer Ausführungsform parallel, an alle anderen Knoten in dem Netzwerk außer an den anfordernden Knoten (Knoten **102** in Transaktion No. 2) und an sich selbst (Knoten **100**). In diesem Fall ist die Anzahl von Nachrichten, die durch die Netzwerkschnittstelle **12** vom Heimatknoten ausgesendet werden, gleich Gesamtzahl -2, wobei Gesamtzahl die Gesamtanzahl von Knoten im Computernetzwerk **10** darstellt.

[0056] Wenn ein Knoten, der die Nachricht vom Heimatknoten **100** empfängt, auf die Anforderung, eine Kopie von Speicherblock **408** an den anfordernden Knoten weiterzuleiten, nicht antworten bzw. reagieren kann, d. h. er keine gültige Kopie von Speicherblock **408** hat, dann ignoriert dieser empfangende Knoten einfach die Anforderung vom Heimatknoten **100**. In dieser Transaktion antwortet Knoten **104**, der gegenwärtig die einzige exklusive Kopie von Speicherblock **408** hat (wie in Zeile B dargestellt), über Pfad **714** dem anfordernden Knoten **102** (Knoten **702** in Transaktion No. 2). Knoten **104** ist der abhängige Knoten in dieser Transaktion, da er den Knoten verkörpert, an dem sich aktuell eine gültige Kopie des angeforderten Speicherblocks befindet. Der Knoten **104** (abhängiger Knoten **712**) stuft seine Kopie von einer exklusiven M-Kopie zu einer ungültigen I-Kopie zurück, da aufgrund der Definition in dem Fall, daß ein Knoten in dem Computernetzwerk eine exklusive M-Kopie hat, d. h. der anfordernde Knoten **102**, alle anderen Knoten ungültige I-Kopien haben müssen.

[0057] Wenn der anfordernde Knoten **702** (Knoten **102** in Transaktion No. 2) eine Kopie von Speicherblock **408** empfängt, vermerkt er intern, daß er nun eine exklusive M-Kopie (Zeile D, Spalte 102) hat und bestätigt den Empfang über Pfad **706**. Wenn der Heimatknoten **100** (Knoten **708**) die Bestätigungsnachricht vom anfordernden Knoten über Pfad **706** empfängt, aktualisiert er seine Kopie, falls notwendig (es stellt sich heraus, daß es in diesem Fall nicht nötig

ist), und setzt das Pending-Flag, das dem Speicherblock **408** zugeordnet ist, zurück, so daß andere Transaktionen, die den Speicherblock **408** beinhalten, bedient werden können.

[0058] Wie aus der Diskussion bezüglich Transaktion No. 2 eingeschätzt werden kann, ist dann, wenn der Heimatknoten keine exklusive M-Kopie von einem Speicherblock hat und eine RTO-Anforderung empfängt, die sich auf diesen Speicherblock bezieht, die Anzahl von Nachrichten, die er über die Netzinfrastruktur **12** senden muß, um einen anderen Knoten aufzufordern, das Weiterleiten einer Kopie des angeforderten Speicherblocks an den anfordernden Knoten zu unterstützen, mindestens Gesamtzahl -2. Von den versendeten Nachrichten ist die Anzahl von Nachrichten, die "vergeudet" sind, d. h. von den empfangenden abhängigen Knoten ignoriert werden, gleich Gesamtzahl -3 (da einer der empfangenden Knoten der Knoten ist, der durch Senden einer gültigen Kopie von Speicherblock **408** an den anfordernden Knoten reagieren kann). Daher kann in Transaktion No. 2 die Anzahl von Nachrichten, die vom Heimatknoten über die Netzinfrastruktur **12** gesendet werden müssen, um die RTO-Anforderung zu bedienen, größer sein als in dem Fall des Verzeichnisprotokolls. Für relativ kleine Netzwerke, die eine relativ kleine Anzahl von Knoten haben, kann die Anzahl von vergeudeten bzw. umsonst gesandten Nachrichten jedoch durch die wesentlich einfachere Implementierung des verzeichnislosen Protokolls gerechtfertigt werden.

[0059] Transaktion No. 3 (Zeile D zu Zeile A): In Transaktion No. 3 fordert Knoten **102**, der eine exklusive M-Kopie hatte, an, daß der Inhalt des Speicherblocks **408** in den Heimatknoten **100** zurückgeschrieben wird. Ein Knoten kann aus verschiedenen Gründen verlangen, daß ein Speicherblock, den er zuvor im Cache gespeichert hat, zurückgeschrieben wird, z. B. weil er einen anderen Speicherblock im Cache speichern möchte und keinen Platz in seinem Speichercache hat. Unter Bezug auf **Fig.** 6 sendet der anfordernde Knoten **102** (Knoten **702**) eine Zurückschreib- (Write-Back, WB) Transaktion an die Netzinfrastruktur. Die Netzinfrastruktur leitet diese Anforderung an den Heimatknoten von Speicherblock **408**, d. h. Knoten **100**, über Pfad **704** weiter. Beim Empfang dieser WB-Anforderung setzt der Heimatknoten **100** (Knoten **708**) das Pending-Flag, das Speicherblock **408** zugeordnet ist.

[0060] Obwohl der Heimatknoten **100** nur eine ungültige I-Kopie von Speicherblock **408** hat (wie in Zeile D, Spalte **100** von **Fig.** 6 abgebildet), kann er folgern, daß Knoten **102** die exklusive M-Kopie von Speicherblock **408** haben muß, da Knoten **102** ein Zurückschreiben anfordert. Folglich sendet der Heimatknoten **100** (Knoten **708**) eine Nachricht über Pfad **710** zum abhängigen Knoten **712**, um ihn aufzufordern, zurückzuschreiben. Der abhängige Knoten **712** verkörpert den Knoten, der aktuell die Kopie von Speicherblock **408** hat, was zufällig derselbe Knoten ist wie der anfordernde Knoten **102** in dieser Zurückschreib-Transaktion. Daher können der anforderde Knoten **702** und der abhängige Knoten **712** als eine einzelne Entität bzw. Instanz in dieser Transaktion behandelt werden. Der Knoten **102** (abhängiger Knoten **712**/anfordernder Knoten **702**) sendet daraufhin eine Kopie von Speicherblock **408** über Pfad **706** an den Heimatknoten **100**, wo der Inhalt von Speicherblock **408** in den Heimatknoten **100** (Knoten **708**) geschrieben wird. Sobald der Inhalt von Speicherblock **408** zurückgeschrieben ist, kann das Pending-Flag, das dem Speicherblock **408** zugeordnet ist, zurückgesetzt werden.

[0061] In Transaktion No. 3 gibt es eine mögliche Wettlaufsituation, wenn das Eintreffen der Zurückschreibanforderung von Knoten **102** bei Knoten **100** verzögert wird und in der Zwischenzeit ein anderer Knoten in dem Netzwerk eine Transaktion bezüglich Speicherblock **408** aussendet, die bedient wird, bevor die Zurückschreibanforderung vom anfordernden Knoten **102** beim Heimatknoten **100** ankommt. In dieser Situation kann der Heimatknoten **100** beim Empfang der Transaktionsanforderung, die später ausgesandt wurde, aber früher angekommen ist, fortfahren, diese später ausgesandte Anforderung zu bedienen, wenn er erkennt, daß das Periding-Flag, das dem Speicherblock **408** zugeordnet ist, nicht gesetzt wurde.

[0062] Das Bedienen dieser später ausgesandten Anforderung kann dazu führen, daß die Kopie von Speicherblock **408** vom Knoten **102** wegbewegt wird. Wenn die Zurückschreibanforderung von Knoten **102** schließlich beim Heimatknoten **100** eintrifft und bedient wird, kann es sein, daß der Knoten **102** keine exklusive M-Kopie von Speicherblock **408** mehr besitzt. In Übereinstimmung mit einem Aspekt der vorliegenden Erfindung kann ein Heimatknoten, z. B. Knoten **100**, in dem Fall, daß er eine Zurückschreibanforderung empfängt, einfach über Pfad **710** dem anfordernden/abhängigen Knoten antworten, um den anfordernden/abhängigen Knoten aufzufordern, nur zurückzuschreiben, wenn der anfordernde/abhängige Knoten immer noch eine exklusive M-Kopie des Speicherblocks besitzt. Andernfalls ignoriert der Heimatknoten einfach die empfangene Zurückschreibanforderung. Vorteilhafterweise werden dadurch, daß dem Knoten, der das Zurückschreiben anfordert, auf diese Weise geantwortet wird, mögliche Wettlaufsituationen vorteilhafterweise vermieden.

[0063] Transaktion No. 4 (Zeile D zu Zeile E): Knoten **104** wünscht eine gemeinsam genutzte, nur lesbare Kopie von Speicherblock **408** und sendet eine RTS (Read-to-Share, Lies, um gemeinsam zu nutzen) Anforderung an die Netzinfrastruktur, um eine nur lesbare Kopie von Speicherblock **408** anzufordern. Die Netzinfrastruktur **12** leitet die RTS-Anforderung über Pfad **704** von dem anfordernden Knoten **104** (Knoten **702**) an den Heimatknoten **100** (Knoten **708**) weiter, der veranlaßt, daß das Pending-Flag;

das dem Speicherblock **408** zugeordnet ist, gesetzt wird.

[0064] Der Heimatknoten **100** weiß nur, daß er eine ungültige I-Kopie (Zeile D, Spalte 100) besitzt. Ohne Wissen darüber, welcher Knoten in dem Netzwerk eine gemeinsam genutzte S-Kopie oder eine exklusive M-Kopie hat, muß der Heimatknoten **100** (Knoten **708**) eine Anforderung an alle übrigen Knoten in dem Netzwerk per Rundsendung verteilen, um den Knoten, der eine M-Kopie hat, aufzufordern, sich auf eine S-Kopie zurückzustufen und eine Kopie des angeforderten Speicherblocks **408** an den anfordernden Knoten **104** (Knoten **702**) weiterzuleiten. Die Anzahl von Anforderungen, die vom Heimatknoten **100** (Knoten **708**) über Pfad **710** an die übrigen Knoten in dem Computernetzwerk gesendet werden, ist wieder gleich Gesamtzahl -2 (es gibt keinen Bedarf, eine Anforderung an den Heimatknoten oder den Knoten, der eine Kopie von Speicherblock **408** anfordert, zu senden).

[0065] In diesem Fall stellt Knoten **102** den abhängigen Knoten dar, der zuvor die exklusive M-Kopie (Zeile D, Spalte 102) hatte. Der abhängige Knoten **102** (Knoten **712**) sendet daraufhin eine Kopie von Speicherblock **408** an den anfordernden Knoten **104** (Knoten **702**) und stuft gleichzeitig die Kopie, die er hat, von einer exklusiven M-Kopie auf eine gemeinsam genutzte S-Kopie zurück (von Zeile D, Spalte 102 zu Zeile E, Spalte 102). Beim Empfang einer Kopie von Speicherblock **408** sendet der anfordernde Knoten **104** (Knoten **702**) dann eine Bestätigungsnachricht an den Heimatknoten **100** (Knoten **708**), der das Pendin-Flag zurücksetzt.

[0066] Transaktion No. 5 (Zeile F zu Zeile G): Nach einer Ausführungsform kann der Heimatknoten auch eine gemeinsam genutzte, nur lesbare S-Kopie behalten, wann immer es eine gemeinsam genutzte, nur lesbare S-Kopie irgendwo in einem Knoten gibt. Die gemeinsam genutzte, nur lesbare S-Kopie kann an den Heimatknoten **708** zum Beispiel von dem anfordernden Knoten **702** (nachdem er eine Kopie des angeforderten Speicherblocks vom abhängigen Knoten erhalten hat) zusammen mit der Bestätigungsnachricht in Pfad **706** gesendet werden. Da der Heimatknoten eine gemeinsam genutzte, nur lesbare S-Kopie hat, kann er vorteilhafterweise eine nachfolgende RTS-Anforderung von einem anderen Knoten in dem Computernetzwerk bedienen, ohne einen anderen Knoten auffordern zu müssen, seine Kopie an den anfordernden Knoten weiterzuleiten. Diese Transaktion wird als Transaktion No. 5 dargestellt, wenn die Zustände von Speicherblock **408** sich von jenen der Zeile F zu Zeile G von **Fig.** 6 ändern.

[0067] In Transaktion No. 5 haben die Knoten **100** und **102** gleichzeitig gemeinsam genutzte, nur lesbare S-Kopien von Speicherblock **408** und die Knoten **104** und **106** haben ungültige I-Kopien desselben Speicherblocks. Knoten **104** wünscht nun eine gemeinsam genutzte, nur lesbare S-Kopie von Speicherblock **408** und sendet eine RTS-Anforderung aus, die beim Heimatknoten **100** (Knoten **708**) über Pfad **706** eintrifft und dazu führt, daß das Pending-Flag, das dem Speicherblock **408** zugeordnet ist, gesetzt wird. Da der Heimatknoten **100** (Knoten **708**) bereits eine gemeinsam genutzte S-Kopie besitzt, braucht er nicht eine Kopie von Speicherblock **408** von irgendeinem anderen Knoten in dem Netzwerk anzufordern und kümmert sich in der Tat nicht darum, welche anderen Kopien bei den Knoten des Netzwerks vorliegen. Folglich kann man sich den Heimatknoten **708** und den abhängigen Knoten **712** als dieselbe Instanz bzw. Entität vorstellen, d. h. Knoten **100**, und er kann über Pfad **714** dem anfordernden Knoten **104** (Knoten **702**) mit einer Kopie von Speicherblock **408** antworten. Beim Empfang einer Kopie von Speicherblock **408** bestätigt der anfordernde Knoten **104** (Knoten **702**) den Empfang durch Senden einer Nachricht über Pfad **706** an den Heimatknoten **100** (Heimatknoten **708**/abhängiger Knoten **712**), was dazu führt, daß das Pending-Flag, das dem Speicherblock **408** zugeordnet ist, zurückgesetzt wird.

[0068] Transaktion No. 6 (Zeile G zu Zeile H): In Transaktion No. 6 haben die Knoten **100**, **102** und **104** gemeinsam genutzte, nur lesbare S-Kopien, während der Knoten **106** eine ungültige I-Kopie von Speicherblock **408** hat. Anschließend wünscht Knoten **106** (Knoten **702** in **Fig.** 6) eine exklusive M-Kopie und sendet eine RTO-Transaktion an die Netzinfrastruktur **12**. Die Netzinfrastruktur **12** leitet die RTO-Anforderung an den Heimatknoten von Speicherblock **408**, d. h. Knoten **100**, über Pfad **704**, was dazu führt, daß das Pending-Flag, das dem Speicherblock **408** zugeordnet ist, gesetzt wird. Da der Heimatknoten **100** (Knoten **708**) nur weiß, daß er eine gemeinsam genutzte, nur lesbare S-Kopie hat (Zeile G, Spalte **400**) und es in Übereinstimmung mit dem verzeichnislosen Protokoll kein zentralisiertes Verzeichnis im Heimatknoten für ihn zum Befragen gibt, muß der Heimatknoten **100** (Knoten **708**) Nachrichten an andere Knoten in dem Netzwerk senden, um sie aufzufordern, ihre Kopien von Speicherblock **408** zu ungültigen I-Kopien zurückzustufen, wenn sie zuvor gemeinsam genutzte, nur lesbare S-Kopien hatten: Wiederum ist die Anzahl von Nachrichten, die über Pfad **710** an anderen Knoten in dem Netzwerk gesendet werden müssen, gleich Gesamtzahl -2 (da es für den Heimatknoten **100** keinen Bedarf gibt, eine Nachricht an sich selbst oder an den anfordernden Knoten zu senden).

[0069] Darüber hinaus kann Knoten **100** sich selbst als einen abhängigen Knoten behandeln, da sich eine gültige, wenn auch gemeinsam genutzte Kopie von Speicherblock **408** aktuell in Knoten **100** befindet. Folglich können Heimatknoten **708** und abhängiger Knoten **712** als dieselbe Instanz bzw. Entität, d. h. Knoten **100**, angesehen werden. Der Heimatknoten **400** (Heimatknoten **708**/abhängiger Knoten **712**) sendet daraufhin eine Kopie von Speicherblock **408** über Pfad **714** an den anfordernden Knoten **106**

(Knoten **702**).

[0070] Nach einer Ausführungsform sendet der Heimatknoten **100** (Heimatknoten 708/abhängiger Knoten **712**) an den anfordernden Knoten **106** (Knoten **702**) auch Information bezüglich der Anzahl von abhängigen Knoten, an die er die Anforderung zur Zurückstufung gesendet hat. Diese Information wird vom anfordernden Knoten **106** (Knoten **702**) gehalten. Nach dieser Ausführungsform meldet auch jeder abhängige Knoten, an den der Heimatknoten **708** die Nachricht, sich zurückzustufen, (über Pfad **710**) gesendet hat, an den anfordernden Knoten **106** (Knoten **702**) über Pfad **714**, um zu bestätigen, daß er entweder seine Kopie von einer gemeinsam genutzte S-Kopie zu einer ungültigen I-Kopie zurückgestuft hat, oder daß er bereits eine ungültige I-Kopie hat und nicht weiter zurückzustufen braucht. Der anfordernde Knoten **106** (Knoten **702**) zählt daraufhin die Anzahl von Bestätigungen, um sicherzustellen, daß alle abhängigen Knoten in dem Netzwerk bestätigt haben.

[0071] Sobald der anfordernde Knoten **106** (Knoten **702**) dadurch zufriedengestellt ist, daß alle Knoten, die ihre Kopien zurückstufen müssen, dies getan haben, sendet der anfordernde Knoten **106** (Knoten **702**) eine Bestätigungsmeldung über Pfad **706** an den Heimatknoten **100** (Knoten **708**); um es dem Heimatknoten **100** zu ermöglichen, das Pending-Flag, das denn Speicherblock **408** zugeordnet ist, zurückzusetzen. Wiederum sollte beachtet werden, daß die Verwendung dieser ausdrücklichen bzw. expliziten Nachrichten und der Pending-Flags vorteilhafterweise die Notwendigkeit beseitigt, einen anordnenden Schaltkreis wie zum Beispiel einen Vermittlerschaltkreis in dem Netzwerk zu haben; um eine natürliche Reihenfolge zu ermöglichen bzw. zu unterstützen.

[0072] Wie Fachleute auf diesem Gebiet aus dem Vorstehenden erkennen, hat das verzeichnislose Protokoll einen Vorteil gegenüber dem busbasierten Rundsende-Verfahren, da im schlimmsten Fall die Anzahl von Nachrichten die an abhängige Knoten gesendet werden muß, um zwei kleiner ist als die gesamte Anzahl von Knoten im Netzwerk. Nach einer Ausführungsform können Nachrichten in dem verzeichnislosen Protokoll, wenn sie von einem Heimatknoten an einen abhängigen Knoten oder abhängige Knoten gesendet werden müssen, jeweils einzeln nacheinander an spezifische abhängige Knoten gesendet werden. Diese Ausführungsform steht im Gegensatz zu Protokollen nach Mitlese-Art oder busbasierten Protokollen nach dem Stand der Technik, bei denen ein Rundsenden vom Heimatknoten an alle anderen Knoten über die gemeinsame Netzinfrastruktur typischerweise durchgeführt wird.

[0073] Im besten Fall, z. B. in Transaktion No. 1, ist die Anzahl von Nachrichten, die gesendet werden, so gering wie im Verzeichnisprotokoll benötigt. Insbesondere beseitigt das verzeichnislose Protokoll der Erfindung vorteilhafterweise die Notwendigkeit, eine Anforderung an irgendeinen anderen Knoten im Netzwerk zu senden, wenn der Heimatknoten eine M-Kopie besitzt und auf eine RTOoder eine RTS-Anforderung von einem anderen Knoten antwortet. Darüber hinaus ist keine Anforderung an irgendeinen anderen Knoten im Netzwerk notwendig, wenn der Heimatknoten eine S-Kopie besitzt und auf eine RTS-Anforderung von einem anderen Knoten antwortet.

[0074] Darüber hinaus stellt das verzeichnislose Protokoll eine Verbesserung gegenüber dem Verzeichnisprotokoll dar, indem die Kosten der Implementierung eines Verzeichnisses beseitigt bzw. eliminiert werden einschließlich der Speicherkosten, die mit der Implementierung einer Verzeichnisstruktur zum Speichern der Verzeichniseinträge einhergehen, und der Entwurfs- bzw. Planungskosten, die mit der Implementierung der Verzeichnislogik und der Fehlersuche und -behebung in der Verzeichnislogik einhergehen.

[0075] Während die vorliegende Erfindung im Hinblick auf verschiedene bevorzugte Ausführungsformen beschrieben wurde, gibt es Abänderungen, Permutationen und Äquivalente, die in den Schutzumfang dieser Erfindung fallen. Es sollte darauf hingewiesen werden, daß es viele alternative Arten der Implementierung der Verfahren und der Vorrichtungen der vorliegenden Erfindung gibt. Die folgenden, beigefügten Ansprüche sollen daher in dem Sinne interpretiert werden, daß sie alle solche Abänderungen, Permutationen und Äquivalente als in den tatsächlichen Schutzumfang. der vorliegenden Erfindung fallend umfassen.

[0076] Bestimmte und bevorzugte Aspekte der Erfindung sind in den begleitenden, unabhängigen und abhängigen Ansprüchen dargelegt. Eigenschaften der abhängigen Ansprüche können mit denen der unabhängigen Ansprüche, soweit angemessen und in Kombinationen, die anders sind, als diejenigen, die explizit in den Ansprüchen ausdrücklich dargelegt sind, kombiniert werden.

Patentansprüche

1. Verfahren, welches in einem Computernetzwerk (**10**) ausgeführt wird, das eine erste Mehrzahl von Knoten hat, die mit einer gemeinsamen Netzwerk-Infrastruktur (**12**) verbunden sind, sowie einen verteilten, gemeinsam verwendeten Speicher hat, der über die erste Vielzahl von Knoten (**100**-**106; 702, 708, 712**) hinweg verteilt ist, um eine erste Speicherzugriffsanforderung durch einen ersten Knoten (**702**) des Computernetzwerks zu bearbeiten, wobei das Computernetzwerk keinen natürlichen Ordnungsmechanismus und keine natürliche, allgemeine Aussendung hat, die sich auf einen Speicherblock (**408**) bezieht, der einen Heimatknoten (**708**) hat, welcher sich von dem ersten Knoten in dem Computernetzwerk unterscheidet, wobei der Heimatknoten ein Protokoll ohne Verzeichnis unterstützt, und zwar durch Nachführen nur des Zustandes des Speicherblockes an

dem Heimatknoten, und kein zentrales Verzeichnis für das Nachführen von Zuständen des Speicherblockes in Knoten der Mehrzahl von Knoten, außer dem Heimatknoten, hat, wobei das Verfahren aufweist:
Empfangen (**704**) der ersten Speicherzugriffsanforderung für den Speicherblock an dem Heimatknoten von dem ersten Knoten über die gemeinsame Netzwerk-Infrastruktur (**12**),
Markieren eines Zustandes des Speicherblockes als anhängig, um zu kennzeichnen, daß ein Zugriff auf den Speicherblock in Bearbeitung ist, wobei dieser Zustand, während er anhängig ist, den Heimatknoten unfähig macht, eine andere Speicherzugriffsanforderung zu bearbeiten, die sich auf diesen Speicherblock bezieht,
Senden (**710**), wenn der Heimatknoten keine erste gültige Kopie des Speicherblockes hat, einer Anforderung von dem Heimatknoten an eine zweite Mehrzahl von Knoten in dem Computernetzwerk, um anzufordern, daß ein zweiter Knoten in dem Computernetzwerk eine erste gültige Kopie des Speicherblockes an den ersten Knoten sendet, wobei die zweite Mehrzahl von Knoten die erste Mehrzahl von Knoten mit Ausnahme des ersten Knotens und des Heimatknotens repräsentiert, wobei die erste gültige Kopie des Speicherblockes eine gültige Kopie repräsentiert, die in der Lage ist, die erste Speicherzugriffsanforderung zu bedienen bzw. zu bearbeiten,
nach Empfang einer Bestätigung von dem ersten Knoten, wodurch bestätigt wird, daß die erste Speicherzugriffsanforderung erfüllt worden ist, Markieren des Zustandes des Speicherblockes als nicht anhängig, um zu kennzeichnen, daß keine Zugriffsanforderung, die sich auf den Speicherblock bezieht; in Bearbeitung ist, um dadurch zu ermöglichen, daß andere Speicherzugriffsanforderungen, die sich auf den Speicherblock beziehen, bearbeitet werden.

2. Verfahren nach Anspruch 1, welches den Schritt aufweist, daß, nachdem der zweite Knoten die erste gültige Kopie des Speicherblockes an den ersten Knoten gesendet hat, die erste gültige Kopie des Speicherblockes auf dem zweiten Knoten von einer Kopie eines ersten Typs zu einer Kopie eines zweiten Typs herabgestuft wird, welcher sich von dem ersten Typ unterscheidet.

3. Verfahren nach Anspruch 2, wobei die erste Speicherzugriffsanforderung eine Anforderung nach einer exklusiven M-Kopie des Speicherblockes ist, wobei die Kopie des ersten Typs eine exklusive M-Kopie repräsentiert und die Kopie des zweiten Typs eine ungültige I-Kopie repräsentiert.

4. Verfahren nach Anspruch 1, welches weiterhin den Schritt aufweist, daß die erste gültige Kopie des Speicherblockes von dem Heimatknoten an den ersten Knoten gesendet wird, wenn der Heimatknoten die erste gültige Kopie des Speicherblockes besitzt.

5. Verfahren nach Anspruch 4, wobei die erste Speicherzugriffsanforderung eine Anforderung nach einer geteilten bzw. gemeinsam verwendeten S-Kopie des Speicherblockes repräsentiert und die erste gültige Kopie des Speicherblockes eine geteilte S-Kopie des Speicherblockes repräsentiert.

6. Verfahren nach Anspruch 4, wobei die erste, Speicherzugriffsanforderung eine Anforderung nach einer exklusiven M-Kopie des Speicherblockes repräsentiert und die erste gültige Kopie des Speicherblockes eine M-Kopie des Speicherblockes repräsentiert,, wobei das Verfahren weiterhin den Schritt aufweist, daß, nachdem der Heimatknoten die erste gültige Kopie des Speicherblockes von dem Heimatknoten an den ersten Knoten gesendet hat, die erste gültige Kopie des Speicherblockes an dem Heimatknoten von einer M-Kopie des Speicherblockes zu einer ungültigen I-Kopie des Speicherblockes herabgestuft wird.

7. Vorrichtung, um einem ersten Knoten (**100**) eines Computernetzwerkes (**10**) einen Zugriff auf eine Kopie eines Speicherblockes (**408**) eines verteilten, gemeinsam verwendeten Speichers des Computernetzwerkes zu gewähren, welches eine erste Mehrzahl von Knoten (**100**-**106**; **702**, **708**, **712**) hat, wobei der Speicherblock einen Heimatknoten (**708**) hat, der sich von dem ersten Knoten (**702**) in dem Computernetzwerk unterscheidet, wobei das Computernetzwerk keinen natürlichen Ordnungsmechanismus und keine natürliche, allgemeine Aussendung hat, wobei der Heimatknoten ein verzeichnisfreies Protokoll unterstützt, indem er so betreibbar ist, daß er den Zustand des Speicherblockes an dem Heimatknoten verfolgt, jedoch kein zentrales Verzeichnis für das Verfolgen von Zuständen des Speicherblockes in Knoten der ersten Mehrzahl von Knoten mit Ausnahme des Heimatknotens hat, wobei die Vorrichtung aufweist:
eine Einrichtung (**704**), um über eine gemeinsame Netzwerk-Infrastruktur (**12**) des Computernetzwerkes an dem Heimatknoten von dem ersten Knoten eine erste Speicherzugriffsanforderung nach dem Speicherblock zu erhalten, und
Einrichtungen, um einen Zustand des Speicherblockes als anhängig herzustellen, um anzuzeigen, daß ein Zugriff auf diesen Speicherblock in Bearbeitung ist, wobei der Zustand während des Anhängigseins den Heimatknoten unfähig macht, andere Speicherzugriffsanforderungen zu bearbeiten, die sich auf den Speicherblock beziehen, und
Einrichtungen, um eine Anforderung von dein Heimatknoten an eine zweite Mehrzahl von Knoten in dem Computernetzwerk zu senden (**710**), wenn der Heimatknoten keine erste gültige Kopie des Speicherblockes hat, um anzufordern, daß ein zweiter Knoten in dem Computernetzwerk die erste gültige Kopie des Speicherblockes an den ersten Knoten sendet, wobei die zweite Mehrzahl von Knoten die

erste Mehrzahl von Knoten repräsentiert mit Ausnahme des ersten Knotens und des Heimatknotens, wobei die erste gültige Kopie des Speicherblockes eine zweite gültige Kopie anfordert, die in der Lage ist, die erste Speicherzugriffsanforderung zu bearbeiten, und

Einrichtungen; um nach dem Empfang einer Bestätigung von dem ersten Knoten, welche bestätigt, daß die erste Speicherzugriffsanforderung erfüllt worden ist; den Status des Speicherblokkes als nicht mehr anhängig zu markieren, um anzuzeigen, daß keine Zugriffsanforderung, die sich auf den Speicherblock bezieht, in Bearbeitung ist, um dadurch zu erlauben, daß die anderen Speicherzugriffsanforderungen, welche sich auf den Speicherblock beziehen, bearbeitet werden.

8. Vorrichtung nach Anspruch 7, welche weiterhin eine Struktur einer Anhängigkeitsanzeige (Flagge) aufweist, welche Struktureinträge für die Anhängigkeit hat, um den Zustand zu bewahren, wobei die Struktur für die Anhängigkeitsanzeige in einem Computerspeicher implementiert ist, der zu dem Heimatknoten gehört.

9. Vorrichtung nach Anspruch 8, wobei die Anzahl der Einträge der Anhängigkeitsstruktur in der Struktur der Anhängigkeitsanzeige gleich der Anzahl von Speicherblöcken in einem Speichermodul an dem Heimatknoten ist, wobei jeder der Einträge der Anhängigkeitsstruktur ein einzelnes Bit aufweist.

10. Vorrichtung nach Anspruch 8, wobei die Anzahl der Einträge der Anhängigkeitsstruktur in der Struktur der Anhängigkeitsanzeige geringer ist als die Anzahl von Speicherblöcken in einem Speichermodul an dem Heimatknoten, wobei jeder der Einträge der Anhängigkeitsstruktur ein Adreßfeld und ein Feld einer Gültigkeitsanzeige aufweist.

11. Vorrichtung nach Anspruch 8, wobei die zweite gültige Kopie des Speicherblockes in dem zweiten Knoten, nachdem der zweite Knoten die zweite gültige Kopie des Speicherblockes von einem Speichercache des zweiten Knotens an den ersten Knotengesendet hat, von einer Kopie eines ersten Typs zu einer Kopie eines zweiten Typs herabgestuft wird, der sich von dem ersten Typ unterscheidet.

12. Vorrichtung nach Anspruch 11, wobei die erste Speicherzugriffsanforderung eine Anforderung nach einer exklusiven M-Kopie des Speicherblockes ist, wobei die Kopie des ersten Typs eine exklusive M-Kopie repräsentiert, und die Kopie des zweiten Typs eine ungültige I-Kopie repräsentiert.

13. Vorrichtung nach Anspruch 8, welcher weiterhin eine Einrichtung aufweist, um die erste gültige Kopie des Speicherblockes von dem Heimatknoten an den ersten Knoten zu senden, wenn der Heimatknoten die erste gültige Kopie des Speicherblockes besitzt.

14. Vorrichtung nach Anspruch 13, wobei die Speicherzugriffsanforderung eine Anforderung nach einer geteilten bzw. gemeinsam verwendeten S-Kopie des Speicherblockes repräsentiert und die erste gültige Kopie des Speicherblockes eine geteilte bzw. gemeinsam verwendete S-Kopie des Speicherblockes repräsentiert.

15. Vorrichtung nach Anspruch 14, wobei die Speicherzugriffsanforderung eine Anforderung nach einer exklusiven M-Kopie des Speicherblockes repräsentiert und die erste gültige Kopie des Speicherblockes eine M-Kopie des Speicherblockes repräsentiert, wobei das Verfahren weiterhin den Schritt aufweist, daß, nachdem der Heimatknoten die erste gültige Kopie des Speicherblockes von dem Heimatknoten an den ersten Knoten gesendet hat, die erste gültige Kopie des Speicherblockes an dem Heimatknoten von einer M-Kopie des Speicherblockes zu einer ungültigen I-Kopie des Speicherblockes herabgestuft wird.

Es folgen 5 Blatt Zeichnungen

Anhängende Zeichnungen

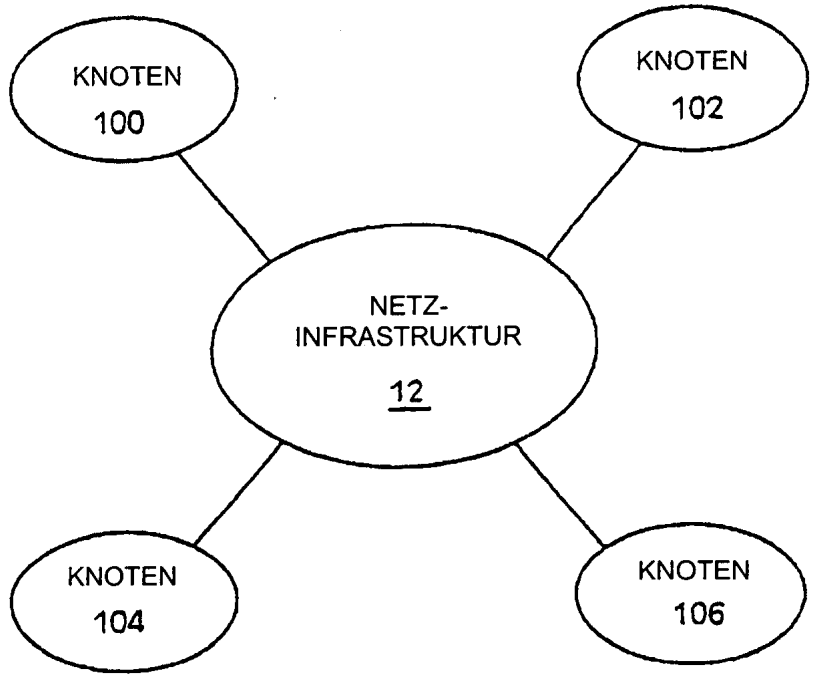


FIG. 1

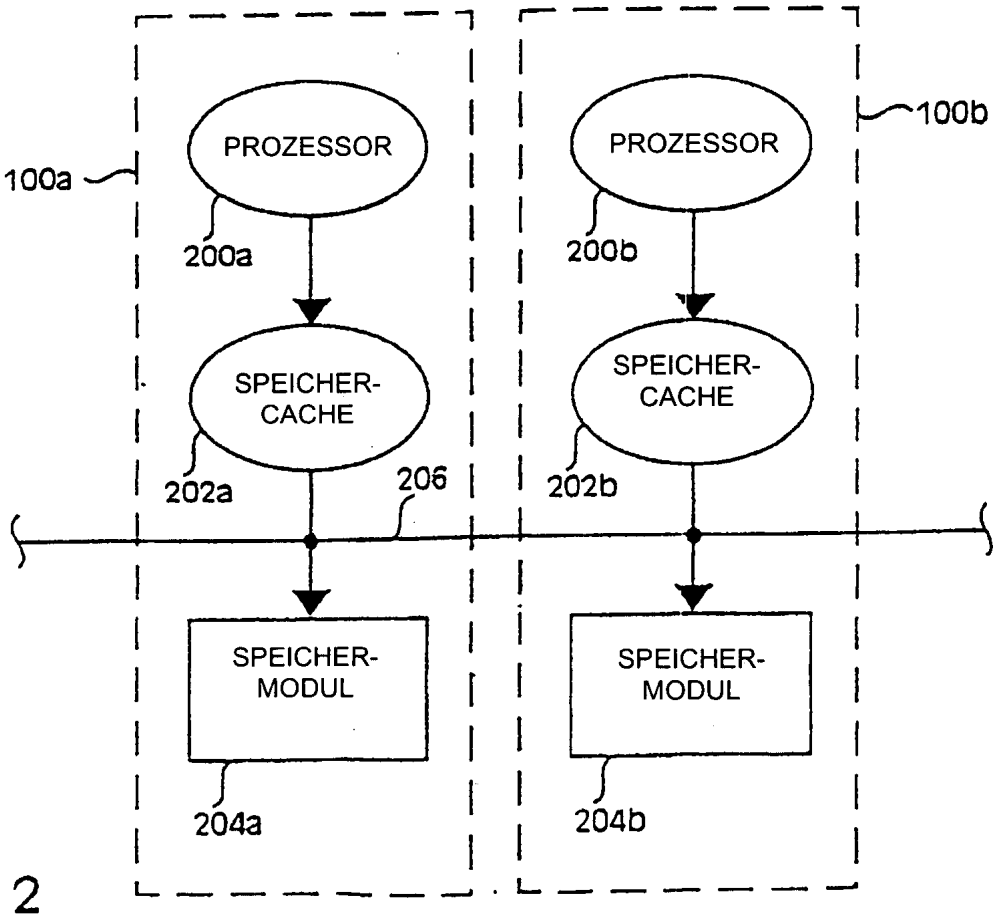


FIG. 2

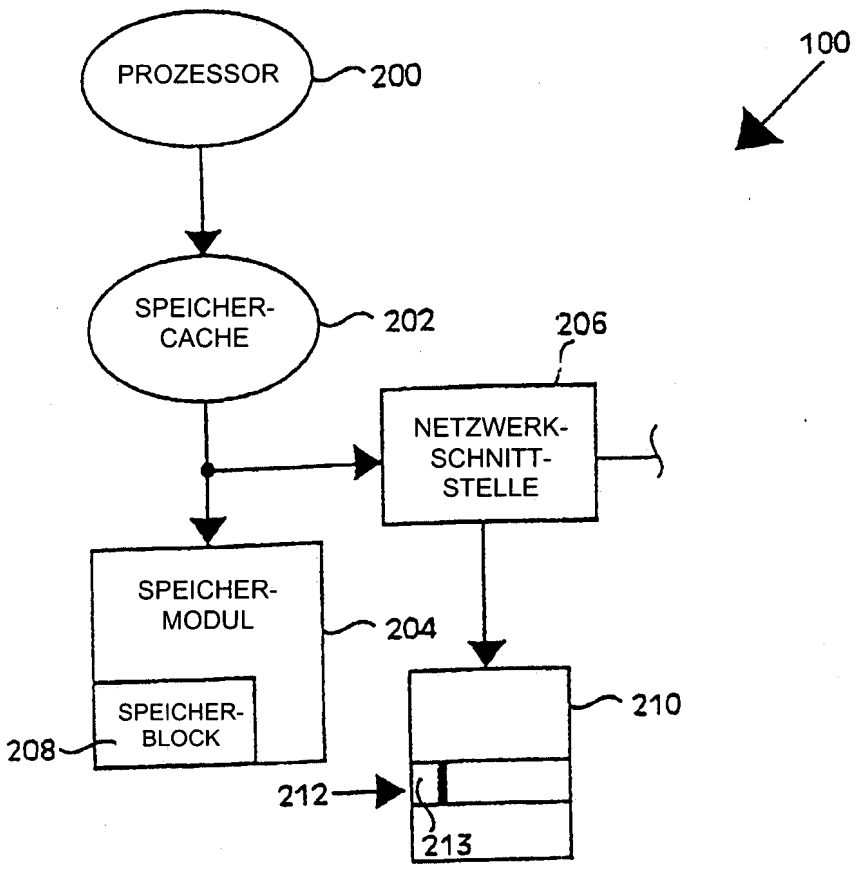

PROZESSOR
200
100
SPEICHER-
CACHE
202
206
NETZWERK-
SCHNITT-
STELLE
SPEICHER-
MODUL
204
SPEICHER-
BLOCK
208
210
212
213

FIG. 3A

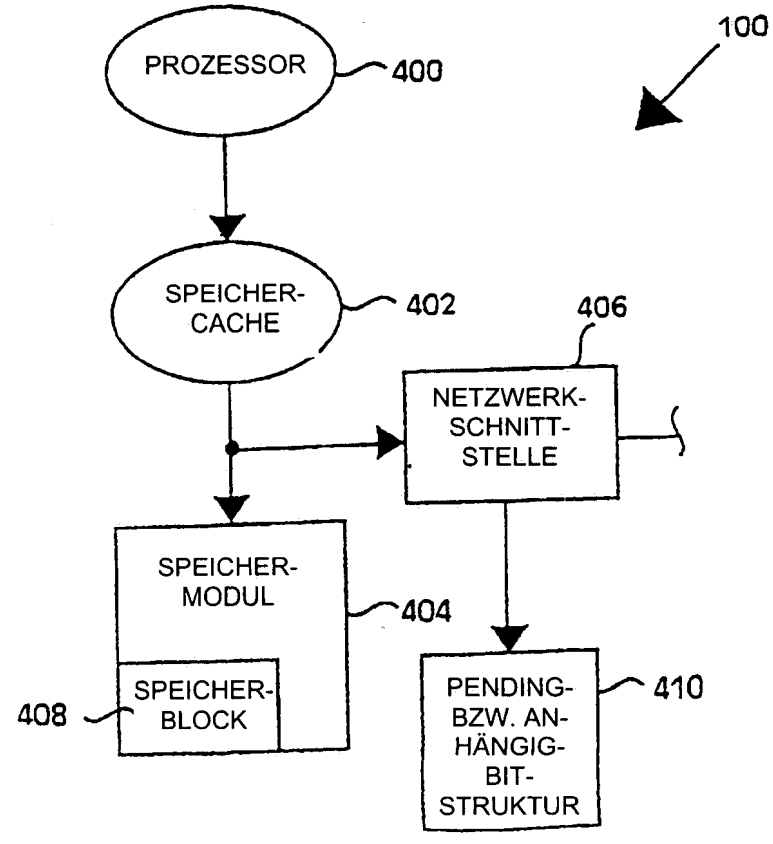

PROZESSOR
400
100
SPEICHER-
CACHE
402
406
NETZWERK-
SCHNITT-
STELLE
SPEICHER-
MODUL
404
SPEICHER-
BLOCK
408
PENDING-
BZW. AN-
HÄNGIG-
BIT-
STRUKTUR
410

Fig. 4

FIG. 3B

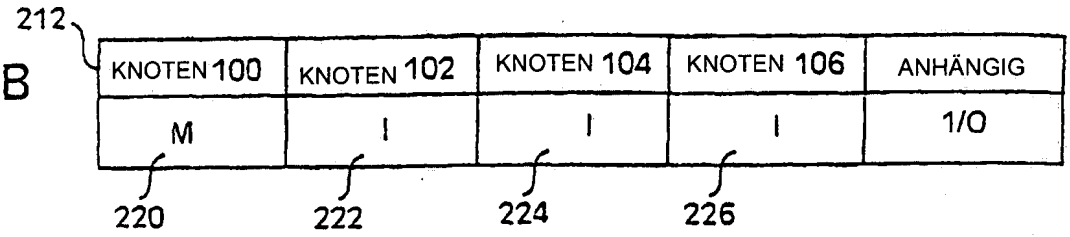


FIG. 3C

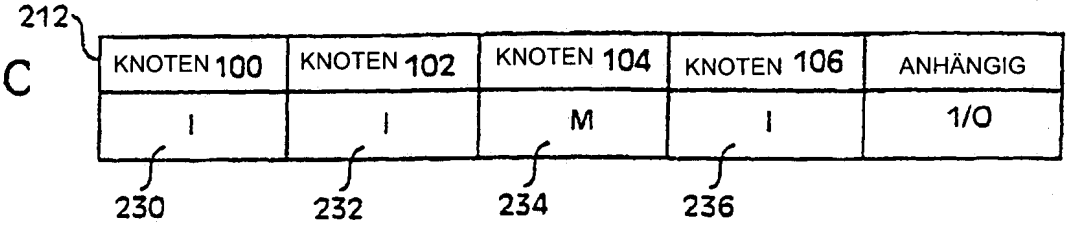
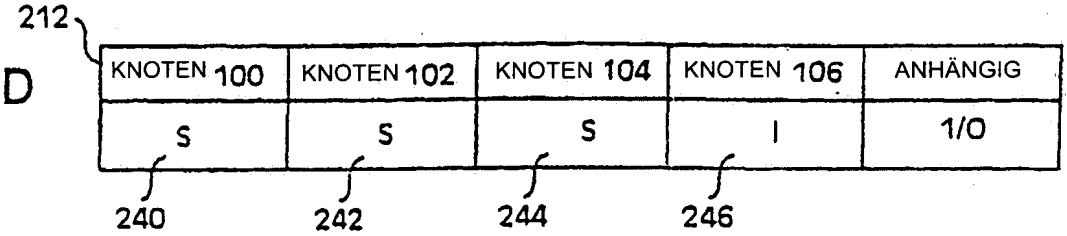


FIG. 3D



ZEILE	KNOTEN 100	KNOTEN 102	KNOTEN 104	KNOTEN 106
A	M	I	I	I
B	I	I	M	I
C	S	S	I	S
D	I	M	I	I
E	S	S	S	I
F	S	S	I	I
G	S	S	S	I
H	I	I	I	M

FIG. 6

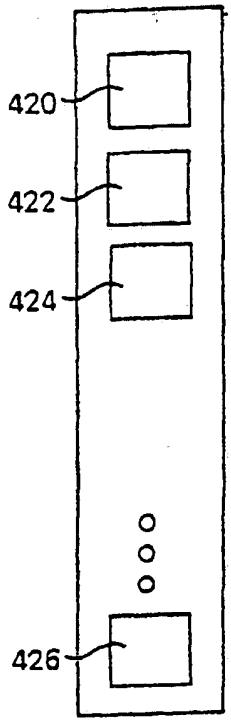

420
422
424
426

FIG. 5A

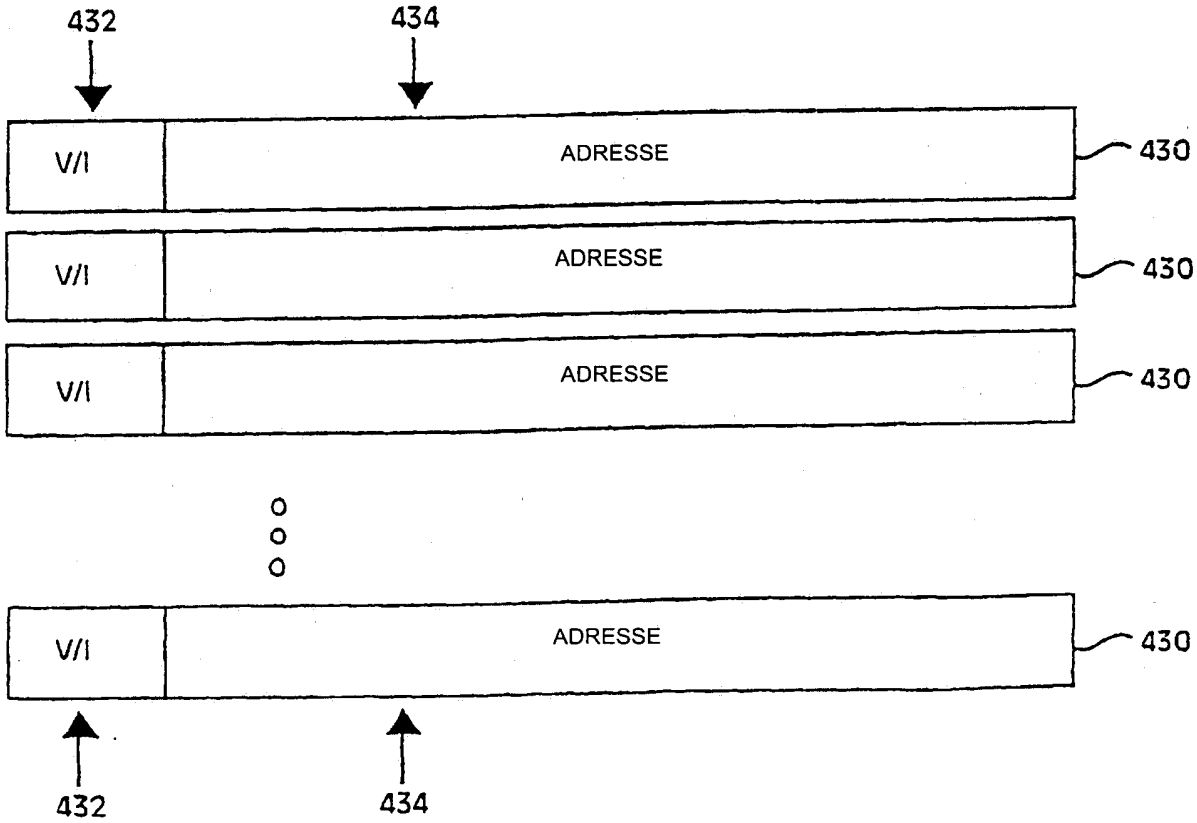

432
434
V/I
ADRESSE
430
V/I
ADRESSE
430
V/I
ADRESSE
430
V/I
ADRESSE
430
432
434

Fig. 5B

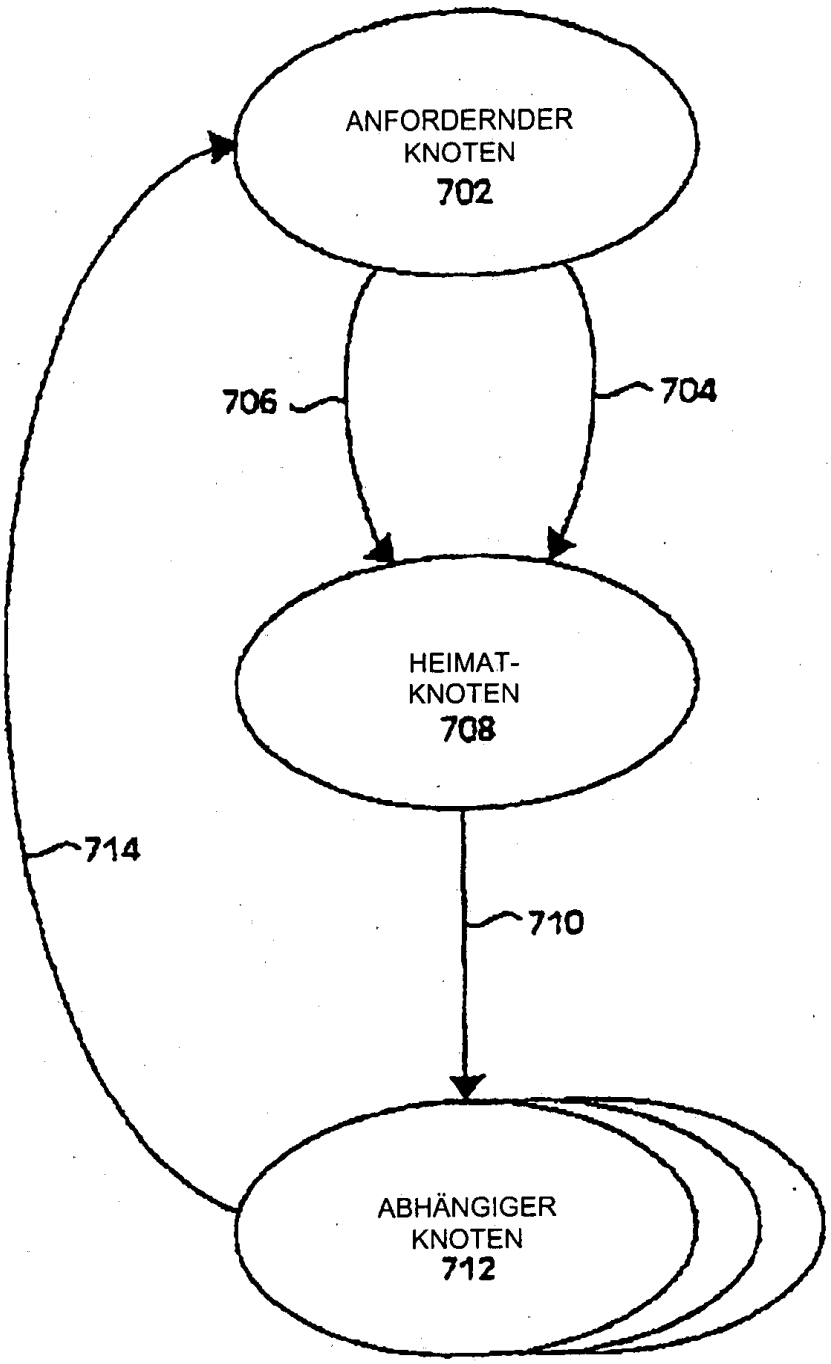

ANFORDERNDER
KNOTEN
702
706
704
HEIMAT-
KNOTEN
708
714
710
ABHÄNGIGER
KNOTEN
712

FIG. 7