(19) Deutsches Patent- und Markenamt

(10) **DE 11 2018 002 049 T5** 2020.01.09

(12) # Veröffentlichung

der internationalen Anmeldung mit der
(87) Veröffentlichungs-Nr.: **WO 2019/012375**
in der deutschen Übersetzung (Art. III § 8 Abs. 2 IntPatÜG)
(21) Deutsches Aktenzeichen: **11 2018 002 049.5**
(86) PCT-Aktenzeichen: **PCT/IB2018/054942**
(86) PCT-Anmeldetag: **04.07.2018**
(87) PCT-Veröffentlichungstag: **17.01.2019**
(43) Veröffentlichungstag der PCT Anmeldung in deutscher Übersetzung: **09.01.2020**

(51) Int Cl.: ***G06F 15/16*** (2006.01)

(30) Unionspriorität:

15/647,561	**12.07.2017**	**US**
15/840,600	**13.12.2017**	**US**

(71) Anmelder:
International Business Machines Corporation, Armonk, N.Y., US

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

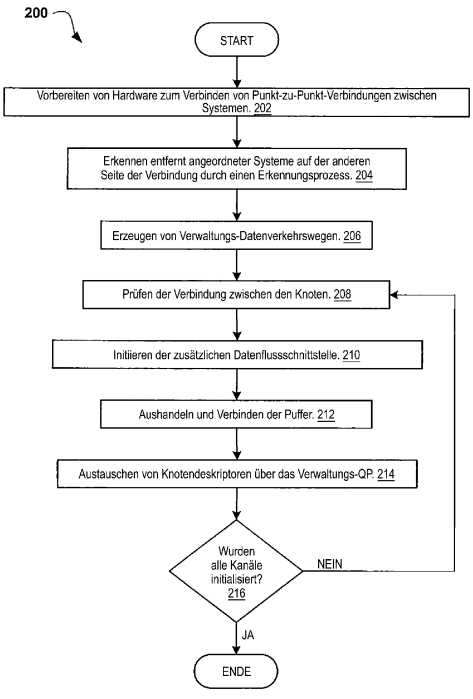
(72) Erfinder:
Lyons, Michael Peter, Cambridge, MA, US; Szwed, Peter Kenneth, Poughkeepsie, N.Y., US; Errickson, Richard, Poughkeepsie, NY, US; Liu, Sean, Poughkeepsie, NY, US

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **VERFAHREN ZUM ERKENNEN ENTFERNT ANGEORDNETER KNOTEN UND ZUM ÜBERPRÜFEN SOWIE VERBINDEN VON ÜBERTRAGUNGSKANÄLEN**

(57) Zusammenfassung: Gemäß einer Ausführungsform werden ein Verfahren, ein Computersystem und ein Computerprogrammprodukt zum Vernetzen von Computern bereitgestellt. Die vorliegende Erfindung kann ein Verfahren zum Verwenden von MAC-Multicast-Prozeduren zum Erkennen einer Adressierbarkeit entfernt angeordneter Knoten und von InfiniBand®-Protokollen beinhalten (InfiniBand® und alle auf InfiniBand® beruhenden Warenzeichen und Logos sind Warenzeichen oder eingetragene Warenzeichen von System I/O, Incorporated und/oder seinen angegliederten Unternehmen), um beim Implementieren von einem oder mehreren Übertragungskanälen Sätze von Warteschlangenpaaren zu verbinden, wobei das Verfahren aufweist: Vorbereiten von Hardware zum Verbinden von Punkt-zu-Punkt-Verbindungen zwischen Systemen; Erkennen von entfernt angeordneten Systemen auf der anderen Seite der Verbindungen durch einen Erkennungsprozess; Erzeugen von Verwaltungs-Datenverkehrswegen; Prüfen der Verbindung zwischen Knoten; Initiieren der zusätzlichen Datenflussschnittstelle; Aushandeln und Verbinden von Puffern; und Austauschen von Knotendeskriptoren.



Beschreibung

HINTERGRUND

[0001] Die vorliegende Erfindung betrifft das Gebiet der Datenverarbeitung im Allgemeinen und insbesondere das Gebiet des Vernetzens von Computern.

[0002] Vernetzen von Computern beinhaltet Verbinden von zwei oder mehr Datenverarbeitungseinheiten mit einer Datenverbindung (data link), um den Datenverarbeitungseinheiten ein Austauschen von Daten zu ermöglichen. Computernetzwerke dienen als Grundlage einer riesigen Anzahl von Anwendungen und Diensten, unter anderem Zugang zum Internet, Cloud-Speicher, Drucker, Faxgeräte, eMail, Nachrichtensofortversand, Telefondienst und digitale Medien. Mit der flächendeckenden Verbreitung von Personal-Computer-Einheiten wie etwa Laptops und Mobiltelefonen sind Computernetzwerke zu einem allgegenwärtigen und wesentlichen Element modernen Lebens geworden. Von daher haben Fortschritte bei der Verwaltung und Verbesserung der Computervernetzungstechnologie das Potenzial, der modernen Gesellschaft weitreichende Vorteile zu bieten.

KU RZDARSTELLU NG

[0003] Unter einem Aspekt wird ein durch einen Prozessor implementiertes Verfahren zum Erkennen entfernt angeordneter Knoten und zum Verbinden einer Mehrzahl von Warteschlangenpaaren zum Implementieren von Übertragungskanälen bereitgestellt, wobei das Verfahren aufweist: Vorbereiten einer Mehrzahl von Hardware-Komponenten zum Verbinden von einer oder mehreren Punkt-zu-Punkt-Verbindungen zwischen einer Mehrzahl von Knoten; Erkennen der Mehrzahl von Knoten in der einen oder den mehreren Punkt-zu-Punkt-Verbindungen durch einen Erkennungsprozess; Erzeugen einer Mehrzahl von Verwaltungs-Datenverkehrswege zwischen der Mehrzahl von Knoten; Prüfen einer Mehrzahl von Verbindungen zwischen der Mehrzahl von Knoten; Initiieren von einer oder mehreren zusätzlichen Datenflussschnittstellen in der Mehrzahl von Verbindungen; Aushandeln und Verbinden einer Mehrzahl von Puffern für die Mehrzahl von Verbindungen; und Austauschen einer Mehrzahl von Knotendeskriptoren zwischen der Mehrzahl von Knoten.

[0004] Unter einem anderen Aspekt wird ein Computersystem zum Erkennen entfernt angeordneter Knoten und zum Verbinden einer Mehrzahl von Warteschlangenpaaren zum Implementieren von Übertragungskanälen bereitgestellt, wobei das Computersystem aufweist: einen oder mehrere Prozessoren, einen oder mehrere durch einen Computer lesbare Speicher, ein oder mehrere durch einen Computer lesbare physische Speichermedien und Programmanweisungen, die zum Ausführen durch mindestens einen der ein oder mehreren Prozessoren über mindestens einen der ein oder mehreren Speicher in mindestens einem der ein oder mehreren physischen Speichermedien gespeichert sind, wobei das Computersystem ein Verfahren ausführen kann, das aufweist: Vorbereiten von Hardware zum Verbinden von einer oder mehreren Punkt-zu-Punkt-Verbindungen zwischen einer Mehrzahl von Knoten; Erkennen der Mehrzahl von Knoten in der einen oder den mehreren Punkt-zu-Punkt-Verbindungen durch einen Erkennungsprozess; Erzeugen einer Mehrzahl von Verwaltungs-Datenverkehrswege zwischen der Mehrzahl von Knoten; Prüfen einer Mehrzahl von Verbindungen zwischen der Mehrzahl von Knoten; Initiieren von einer oder mehreren zusätzlichen Datenflussschnittstellen in der Mehrzahl von Verbindungen; Aushandeln und Verbinden einer Mehrzahl von Puffern für die Mehrzahl von Verbindungen; und Austauschen einer Mehrzahl von Knotendeskriptoren zwischen der Mehrzahl von Knoten.

[0005] Unter einem anderen Aspekt wird ein Computerprogrammprodukt zum Erkennen entfernt angeordneter Knoten und zum Verbinden einer Mehrzahl von Warteschlangenpaaren zum Implementieren von Übertragungskanälen bereitgestellt, wobei das Computerprogrammprodukt aufweist: ein oder mehrere durch einen Computer lesbare physische Speichermedien und Programmanweisungen, die in mindestens einem der ein oder mehreren physischen Speichermedien gespeichert sind, wobei die Programmanweisungen durch einen Prozessor ausgeführt werden können, wobei die Programmanweisungen den Prozessor steuern, so dass er ein Verfahren ausführt, das aufweist: Vorbereiten einer Mehrzahl von Hardware-Komponenten zum Verbinden von einer oder mehreren Punkt-zu-Punkt-Verbindungen zwischen einer Mehrzahl von Knoten; Erkennen der Mehrzahl von Knoten in der einen oder den mehreren Punkt-zu-Punkt-Verbindungen durch einen Erkennungsprozess; Erzeugen einer Mehrzahl von Verwaltungs-Datenverkehrswege zwischen der Mehrzahl von Knoten; Prüfen einer Mehrzahl von Verbindungen zwischen der Mehrzahl von Knoten; Initiieren von einer oder mehreren zusätzlichen Datenflussschnittstellen in der Mehrzahl von Verbindungen; Aushandeln und Verbinden einer Mehrzahl von Puffern für die Mehrzahl von Verbindungen; und Austauschen einer Mehrzahl von Knotendeskriptoren zwischen der Mehrzahl von Knoten.

[0006] Gemäß einer Ausführungsform werden ein Verfahren, ein Computersystem und ein Computerprogrammprodukt zum Vernetzen von Computern bereitgestellt. Die vorliegende Erfindung kann ein Verfahren des Verwendens von MAC-Multicast-Prozeduren (Media Access Control (MAC) multicast procedures) zum Erkennen einer Adressierbarkeit entfernt angeordneter Knoten und InfiniBand®-Protokolle enthalten (InfiniBand® und alle auf InfiniBand®

beruhende Warenzeichen und Logos sind Warenzeichen oder eingetragene Warenzeichen von System I/O, Incorporated und/oder seinen angegliederten Unternehmen), um Sätze von Warteschlangenpaaren beim Implementieren von einem oder mehreren Übertragungskanälen zu verbinden, wobei das Verfahren aufweist: Vorbereiten von Hardware zum Verbinden von Punkt-zu-Punkt-Verbindungen zwischen Systemen; Erkennen von entfernt angeordneten Systemen auf der anderen Seite der Verbindungen durch einen Erkennungsprozess; Erzeugen von Verwaltungs-Datenverkehrswege; Prüfen der Verbindung zwischen Knoten; Initiieren der zusätzlichen Datenflussschnittstelle; Aushandeln und Verbinden von Puffern; und Austauschen von Knotendeskriptoren.

Figurenliste

[0007] Bevorzugte Ausführungsformen der vorliegenden Erfindung werden nun lediglich beispielhaft unter Bezugnahme auf die folgenden Zeichnungen beschrieben:

Fig. 1 veranschaulicht eine beispielhafte vernetzte Computerumgebung gemäß mindestens einer Ausführungsform;

Fig. 2 ist ein Funktionsablaufplan, der einen Prozess zum Erkennen und Zuteilen von Knoten gemäß mindestens einer Ausführungsform veranschaulicht;

Fig. 3 ist ein Funktionsschaubild, das eine Prozedur zum Herstellen von Verbindungen gemäß mindestens einer Ausführungsform veranschaulicht;

Fig. 4 ist ein Blockschaltbild von internen und externen Komponenten von in **Fig. 1** gezeigten Computern und Servern gemäß mindestens einer Ausführungsform;

Fig. 5 zeigt eine Cloud-Computing-Umgebung gemäß einer Ausführungsform der vorliegenden Erfindung; und

Fig. 6 zeigt Abstraktionsmodellschichten (abstraction model layers) gemäß einer Ausführungsform der vorliegenden Erfindung.

[0008] Die verschiedenen Merkmale der Zeichnungen sind nicht maßstabsgetreu, da die Abbildungen dazu dienen sollen, Fachleuten ein Verständnis bevorzugter Ausführungsformen der Erfindung in Verbindung mit der ausführlichen Beschreibung zu ermöglichen.

[0009] Ausführliche Ausführungsformen der beanspruchten Strukturen und Verfahren werden hierin beschrieben; es versteht sich jedoch, dass die offenbarten Ausführungsformen lediglich die beanspruchten Strukturen und Verfahren veranschaulichen, die in verschiedenen Formen ausgeführt werden können. Diese Erfindung kann jedoch in vielen verschiedenen Formen ausgeführt werden und soll nicht so ausgelegt werden, dass sie auf die hierin dargelegten beispielhaften Ausführungsformen beschränkt ist. In der Beschreibung können Einzelheiten allgemein bekannter Merkmale und Techniken weggelassen werden, damit die dargestellten Ausführungsformen nicht unnötig in den Hintergrund treten.

[0010] Ausführungsformen der vorliegenden Erfindung betreffen das Gebiet der Datenverarbeitung und insbesondere Vernetzen von Computern. Die folgenden beschriebenen beispielhaften Ausführungsformen stellen ein System, ein Verfahren und ein Programmprodukt bereit, um unter anderem MAC-Multicast-Prozeduren zum Erkennen einer Adressierbarkeit entfernt angeordneter Knoten und InfiniBand®-Protokolle (InfiniBand® und alle auf InfiniBand® beruhenden Warenzeichen und Logos sind Warenzeichen oder eingetragene Warenzeichen von System I/O, Incorporated und/oder seinen angegliederten Unternehmen) zum Verbinden von Warteschlangenpaaren beim Implementieren von einem oder mehreren Übertragungskanälen zu verwenden. Daher hat die vorliegende Ausführungsform die Fähigkeit, das technische Gebiet des Vernetzens von Computern zu verbessern, indem sie ein Inband-Verfahren zum Erkennen von Knoten bereitstellt, das tolerant gegenüber Vermittlungsumgebungen ist.

[0011] Wie zuvor beschrieben, beinhaltet Vernetzen von Computern das Verbinden von zwei oder mehreren Datenverarbeitungseinheiten mit einer Datenverbindung, um den Datenverarbeitungseinheiten zu ermöglichen, Daten auszutauschen. Computernetzwerke dienen als Grundlage einer riesigen Anzahl von Anwendungen und Diensten, unter anderem Zugang zum Internet, Cloud-Speicher, Drucker, Faxgeräte, eMail, Nachrichtensofortversand, Telefondienst und digitale Medien. Mit der flächendeckenden Verbreitung von Personal-Computer-Einheiten wie etwa Laptops und Mobiltelefonen sind Computernetzwerke zu einem allgegenwärtigen und wesentlichen Element modernen Lebens geworden. Von daher haben Fortschritte bei der Verwaltung und Verbesserung der Computervernetzungstechnologie das Potenzial, der modernen Gesellschaft weitreichende Vorteile zu bieten.

[0012] Viele Netzwerkverbindungen verwenden einen Standardsatz von Protokollen, die in handelsübliche Betriebssysteme integriert sind. Diese Protokolle sind allgemeingültig und bilden die Grundlage für die Verbreitung einer Netzwerk-Anschlussfähigkeit. Eine solche Allgemeingültigkeit geht jedoch auf Kosten der Leistungsfähigkeit von Netzwerken. Bei Systemen, die empfindlich gegenüber Messgrößen wie Netzwerk-Latenzzeit sind, kann es erforderlich sein, spezifische Übertragungsprotokolle zu ent-

wickeln, die nicht allgemeingültig, sondern an eine spezifische Anwendung und Betriebsumgebung des Systems angepasst sind. Daher besteht ein Bedarf an einem Verfahren, das Erkennen von Knoten in Punkt-zu-Punkt-Verbindungen ohne die Notwendigkeit eines zentralen Servers verwendet und Daten auf der Grundlage fester Kanäle überträgt, die über anfängliches Erkennen eingerichtet werden, anstatt Kanäle zu verwenden, die unter Verwenden eines dynamischen Protokolls erzeugt werden.

[0013] Gemäß einer Ausführungsform wird ein Satz von einem oder mehreren Kanälen zwischen zwei Systemen eingerichtet durch Inband-Erkennen einer Fern-Adressierbarkeit unter Verwenden eines Multicast-Protokolls, anschließendes Einrichten eines direkt verbundenen Verwaltungspfades, über den Definitionen für entfernt angerordnete Kanäle überprüft und initialisiert werden.

[0014] In einer solchen Ausführungsform wird zuerst unter Verwenden eines Bitübertragungsschicht-Protokolls wie etwa Ethernet eine physische Verbindung zwischen zwei Serversystemen eingerichtet, wobei ein Kommunikationsadapter verwendet wird, der ein Transportschicht-Protokoll wie etwa InfiniBand® unterstützt (InfiniBand® und alle auf InfiniBand® beruhenden Warenzeichen und Logos sind Warenzeichen oder eingetragene Warenzeichen von System I/O, Incorporated und/oder seinen angegliederten Unternehmen). Über diese Verbindung wird die Adressierbarkeit des entfernt angeordneten Systems erhalten, indem ein lokales Multicast-Warteschlangenpaar (queue pair, QP) eingerichtet wird, um eine Multicast-Nachricht zu empfangen, die die Adresse des verbundenen Anschlusses im entfernt angeordneten System enthält, und um eine Multicast-Nachricht zu übertragen, die die Adresse des verbundenen Anschlusses im lokalen System enthält.

[0015] Wenn die Adresse des entfernt angeordneten Anschlusses erhalten wurde, wird sie als Parameter zum Einrichten eines direkt verbundenen Verwaltungs-QP verwendet. Das Verwaltungs-QP wird als Pfad verwendet, über den Daten übertragen werden, beispielsweise lokale und entfernte Definitionen der Kanäle, und es werden Parameter ausgehandelt wie etwa die Anzahl von QPs, die einen Kanal aufweisen.

[0016] Für jeden mit einer übereinstimmenden lokalen und entfernten Definition wird der Pfad des Verwaltungs-QP verwendet, um ein zusätzliches QP und einen Satz von einzelnen Puffer-QPs zu definieren und zu verbinden. Das zusätzliche QP wird für kanalspezifische Nicht-Datennachrichten (wie etwa einen Nachrichtenabbruchbefehl) verwendet, und die Puffer-QPs werden zum Übertragen und Empfangen von Datennachrichten für den Kanal verwendet.

[0017] Bei der vorliegenden Erfindung kann es sich um ein System, ein Verfahren und/oder ein Computerprogrammprodukt mit jedem möglichen technischen Integrationsgrad handeln. Das Computerprogrammprodukt kann (ein) durch einen Computer lesbare(s) Speichermedium (oder -medien) beinhalten, auf dem/denen durch einen Computer lesbare Programmanweisungen gespeichert ist/sind, um einen Prozessor dazu zu veranlassen, Aspekte der vorliegenden Erfindung auszuführen.

[0018] Bei dem durch einen Computer lesbaren Speichermedium kann es sich um eine physische Einheit handeln, die Anweisungen zur Verwendung durch eine Einheit zum Ausführen von Anweisungen behalten und speichern kann. Bei dem durch einen Computer lesbaren Speichermedium kann es sich zum Beispiel um eine elektronische Speichereinheit, eine magnetische Speichereinheit, eine optische Speichereinheit, eine elektromagnetische Speichereinheit, eine Halbleiterspeichereinheit oder jede geeignete Kombination daraus handeln, ohne auf diese beschränkt zu sein. Zu einer nicht erschöpfenden Liste spezifischerer Beispiele des durch einen Computer lesbaren Speichermediums gehören die Folgenden: eine tragbare Computerdiskette, eine Festplatte, ein Direktzugriffsspeicher (RAM), ein Nur-Lese-Speicher (ROM), ein löschbarer programmierbarer Nur-Lese-Speicher (EPROM bzw. Flash-Speicher), ein statischer Direktzugriffsspeicher (SRAM), ein tragbarer Kompaktspeicherplatte-Nur-Lese-Speicher (CD-ROM), eine DVD (digital versatile disc), ein Speicher-Stick, eine Diskette, eine mechanisch codierte Einheit wie zum Beispiel Lochkarten oder gehobene Strukturen in einer Rille, auf denen Anweisungen gespeichert sind, und jede geeignete Kombination daraus. Ein durch einen Computer lesbares Speichermedium soll in der Verwendung hierin nicht als flüchtige Signale an sich aufgefasst werden, wie zum Beispiel Funkwellen oder andere sich frei ausbreitende elektromagnetische Wellen, elektromagnetische Wellen, die sich durch einen Wellenleiter oder ein anderes Übertragungsmedium ausbreiten (z.B. ein Lichtwellenleiterkabel durchlaufende Lichtimpulse) oder durch einen Draht übertragene elektrische Signale.

[0019] Hierin beschriebene, durch einen Computer lesbare Programmanweisungen können von einem durch einen Computer lesbaren Speichermedium auf jeweilige Datenverarbeitungs/Verarbeitungseinheiten oder über ein Netzwerk wie zum Beispiel das Internet, ein lokales Netzwerk, ein Weitverkehrsnetz und/oder ein drahtloses Netzwerk auf einen externen Computer oder eine externe Speichereinheit heruntergeladen werden. Das Netzwerk kann Kupferübertragungskabel, Lichtwellenübertragungsleiter, drahtlose Übertragung, Leitwegrechner, Firewalls, Vermittlungseinheiten, Gateway-Computer und/oder Edge-Server aufweisen. Eine Netzwerkad-

apterkarte oder Netzwerkschnittstelle in jeder Datenverarbeitungs/Verarbeitungs-Einheit empfängt durch einen Computer lesbare Programmanweisungen aus dem Netzwerk und leitet die durch einen Computer lesbaren Programmanweisungen zum Speichern in einem durch einen Computer lesbaren Speichermedium innerhalb der entsprechenden Datenverarbeitungs/Verarbeitungs-Einheit weiter.

[0020] Bei durch einen Computer lesbaren Programmanweisungen zum Ausführen von Arbeitsschritten der vorliegenden Erfindung kann es sich um Assembler-Anweisungen, ISA-Anweisungen (Instruction-Set-Architecture), Maschinenanweisungen, maschinenabhängige Anweisungen, Mikrocode, Firmware-Anweisungen, zustandssetzende Daten oder entweder Quellcode oder Objektcode handeln, die in einer beliebigen Kombination aus einer oder mehreren Programmiersprachen geschrieben werden, darunter objektorientierte Programmiersprachen wie Smalltalk, C++ o.ä. sowie herkömmliche prozedurale Programmiersprachen wie die Programmiersprache „C“ oder ähnliche Programmiersprachen. Die durch einen Computer lesbaren Programmanweisungen können vollständig auf dem Computer des Benutzers, teilweise auf dem Computer des Benutzers, als eigenständiges Software-Paket, teilweise auf dem Computer des Benutzers und teilweise auf einem fernen Computer oder vollständig auf dem fernen Computer oder Server ausgeführt werden. In letzterem Fall kann der entfernt angeordnete Computer mit dem Computer des Benutzers durch eine beliebige Art Netzwerk verbunden sein, darunter ein lokales Netzwerk (LAN) oder ein Weitverkehrsnetz (WAN), oder die Verbindung kann mit einem externen Computer hergestellt werden (zum Beispiel über das Internet unter Verwenden eines Internet-Dienstanbieters). In einigen Ausführungsformen können elektronische Schaltungen, darunter zum Beispiel programmierbare Logikschaltungen, vor Ort programmierbare Gatter-Anordnungen (FPGA, field programmable gate arrays) oder programmierbare Logikanordnungen (PLA, programmable logic arrays) die durch einen Computer lesbaren Programmanweisungen ausführen, indem sie Zustandsinformationen der durch einen Computer lesbaren Programmanweisungen nutzen, um die elektronischen Schaltungen zu personalisieren, um Aspekte der vorliegenden Erfindung durchzuführen.

[0021] Aspekte der vorliegenden Erfindung sind hierin unter Bezugnahme auf Ablaufpläne und/oder Blockschaltbilder bzw. Schaubilder von Verfahren, Vorrichtungen (Systemen) und Computerprogrammprodukten gemäß Ausführungsformen der Erfindung beschrieben. Es versteht sich von selbst, dass jeder Block der Ablaufpläne und/oder der Blockschaltbilder bzw. Schaubilder sowie Kombinationen von Blöcken in den Ablaufplänen und/oder den Blockschaltbildern bzw. Schaubildern mittels durch einen Computer lesbaren Programmanweisungen ausgeführt werden können.

[0022] Diese durch einen Computer lesbaren Programmanweisungen können einem Prozessor eines Universalcomputers, eines Spezialcomputers oder einer anderen programmierbaren Datenverarbeitungsvorrichtung bereitgestellt werden, um eine Maschine zu erzeugen, so dass die über den Prozessor des Computers bzw. der anderen programmierbaren Datenverarbeitungsvorrichtung ausgeführten Anweisungen ein Mittel zur Umsetzung der in dem Block bzw. den Blöcken der Ablaufpläne und/oder der Blockschaltbilder bzw. Schaubilder festgelegten Funktionen/Schritte erzeugen. Diese durch einen Computer lesbaren Programmanweisungen können auch auf einem durch einen Computer lesbaren Speichermedium gespeichert sein, das einen Computer, eine programmierbare Datenverarbeitungsvorrichtung und/oder andere Einheiten so steuern kann, dass sie auf eine bestimmte Art funktionieren, so dass das durch einen Computer lesbare Speichermedium, auf dem Anweisungen gespeichert sind, ein Herstellungsprodukt aufweist, darunter Anweisungen, welche Aspekte der/des in dem Block bzw. den Blöcken des Ablaufplans und/oder der Blockschaltbilder bzw. Schaubilder angegebenen Funktion/Schritts umsetzen.

[0023] Die durch einen Computer lesbaren Programmanweisungen können auch auf einen Computer, eine andere programmierbare Datenverarbeitungsvorrichtung oder eine andere Einheit geladen werden, um das Ausführen einer Reihe von Prozessschritten auf dem Computer bzw. der anderen programmierbaren Vorrichtung oder anderen Einheit zu verursachen, um einen auf einem Computer ausgeführten Prozess zu erzeugen, so dass die auf dem Computer, einer anderen programmierbaren Vorrichtung oder einer anderen Einheit ausgeführten Anweisungen die in dem Block bzw. den Blöcken der Ablaufpläne und/oder der Blockschaltbilder bzw. Schaubilder festgelegten Funktionen/Schritte umsetzen.

[0024] Die Ablaufpläne und die Blockschaltbilder bzw. Schaubilder in den Figuren veranschaulichen die Architektur, die Funktionalität und den Betrieb möglicher Ausführungen von Systemen, Verfahren und Computerprogrammprodukten gemäß verschiedenen Ausführungsformen der vorliegenden Erfindung. In diesem Zusammenhang kann jeder Block in den Ablaufplänen oder Blockschaltbildern bzw. Schaubildern ein Modul, ein Segment oder einen Teil von Anweisungen darstellen, die eine oder mehrere ausführbare Anweisungen zum Ausführen der bestimmten logischen Funktion(en) aufweisen. In einigen alternativen Ausführungen können die in dem Block angegebenen Funktionen in einer anderen Reihenfolge als in den Figuren gezeigt stattfinden. Zwei nacheinander gezeigte Blöcke können zum Beispiel

in Wirklichkeit im Wesentlichen gleichzeitig ausgeführt werden, oder die Blöcke können manchmal je nach entsprechender Funktionalität in umgekehrter Reihenfolge ausgeführt werden. Es ist ferner anzumerken, dass jeder Block der Blockschaltbilder bzw. Schaubilder und/oder der Ablaufpläne sowie Kombinationen aus Blöcken in den Blockschaltbildern bzw. Schaubildern und/oder den Ablaufplänen durch spezielle auf Hardware beruhende Systeme umgesetzt werden können, welche die festgelegten Funktionen oder Schritte durchführen, oder Kombinationen aus Spezial-Hardware und Computeranweisungen ausführen.

[0025] Die im Folgenden beschriebenen beispielhaften Ausführungsformen stellen ein System, ein Verfahren und ein Programmprodukt bereit, um einen Satz von einem oder mehreren Übertragungskanälen zwischen zwei Datenverarbeitungssystemen einzurichten durch Inband-Erkennen einer Fern-Adressierbarkeit unter Verwenden eines Multicast-Protokolls, anschließendes Einrichten eines direkt verbundenen Verwaltungspfades, über den Definitionen entfernt angeordneter Kanäle überprüft und initialisiert werden.

[0026] Unter Bezugnahme auf **Fig. 1** wird eine beispielhafte vernetzte Computerumgebung **100** gemäß mindestens einer Ausführungsform gezeigt. Die vernetzte Computerumgebung **100** kann Netzwerkknoten **102A** und **102B** enthalten, die über ein Datenübertragungsnetz **112** miteinander verbunden sind. Gemäß mindestens einer Ausführung kann die vernetzte Computerumgebung **100** eine Mehrzahl von Netzwerkknoten **102A**, **102B** enthalten, von denen der Übersichtlichkeit halber nur zwei gezeigt werden.

[0027] Das Datenübertagungsnetz **112** kann verschiedene Arten von Datenübertragungsnetzen enthalten, beispielsweise Weitverkehrsnetze (WAN), lokale Netze (LAN), ein Datenfernübertragungsnetz, ein drahtloses Netz, ein öffentliches Netz und/oder ein Satellitennetz. Das Datenübertagungsnetz **112** kann Verbinden enthalten, beispielsweise Leitungen, drahtlose Übertragungsverbindungen oder Lichtwellenleiterverbindungen. Es ist klar, dass **Fig. 1** lediglich eine Veranschaulichung einer Ausführung zeigt und keinerlei Beschränkungen hinsichtlich der Umgebungen, in denen verschiedene Ausführungsformen realisiert werden können, mit sich bringt. An den gezeigten Umgebungen können je nach Anforderungen an Gestaltung und Ausführung viele Modifikationen vorgenommen werden.

[0028] Der Netzwerkknoten **102A** kann einen Prozessor **104A** und eine Datenspeichereinheit **106A** enthalten, die ein Software-Programm **108A** und ein Programm **110A** zum Erkennen und Zuteilen von Knoten (node discovery and arbiter program) aufnehmen und ausführen kann, und kann gemäß einer Ausführungsform der Erfindung über das Datenübertragungsnetz **112** mit dem Netzwerkknoten **102B** Daten austauschen. Bei dem Netzwerkknoten **102A** kann es sich zum Beispiel um eine mobile Einheit, ein Telefon, einen Personal Digital Assistant, ein Netbook, einen Laptop-Computer, einen Tablet-Computer, einen Tisch-Computer oder einen beliebigen Typ von Datenverarbeitungseinheit handeln, die ein Programm ausführen und auf ein Netzwerk zugreifen kann. Wie unter Bezugnahme auf **Fig. 5** erläutert wird, kann der Netzwerkknoten **102A** interne Komponenten **502a** bzw. externe Komponenten **504a** enthalten. Der Netzwerkknoten **102A** kann auch in einem Cloud-Computing-Dienstmodell betrieben werden, beispielsweise Software als Dienstleistung (Software as a Service, SaaS), Plattform als Dienstleistung (Platform as a Service, PaaS) oder Infrastruktur als Dienstleistung (Infrastructure as a Service, IaaS). Der Netzwerkknoten **102A** kann sich auch in einem Cloud-Computing-Einsatzmodell befinden, beispielsweise in einer privaten Cloud (private cloud), einer gemeinschaftlichen Cloud (community cloud), einer öffentlichen Cloud (public cloud) oder einer kombinierten Cloud (hybrid cloud).

[0029] Der Netzwerkknoten **102B** kann einen kann einen Prozessor **104B** und eine Datenspeichereinheit **106B** enthalten, die ein Software-Programm **108B** und ein Programm **110B** zum Erkennen und Zuteilen von Knoten aufnehmen und ausführen kann, und kann gemäß einer Ausführungsform der Erfindung über das Datenübertragungsnetz **112** mit dem Netzwerkknoten **102A** Daten austauschen. Bei dem Netzwerkknoten **102B** kann es sich zum Beispiel um eine mobile Einheit, ein Telefon, ein Personal Digital Assistant, ein Netbook, einen Laptop-Computer, einen Tablet-Computer, einen Desktop-Computer oder einen beliebigen Typ von Datenverarbeitungseinheit handeln, die ein Programm ausführen und auf ein Netzwerk zugreifen kann. Wie unter Bezugnahme auf **Fig. 5** erläutert wird, kann der Netzwerkknoten **102B** interne Knoten **502a** bzw. externe Komponenten **504a** enthalten. Der Netzwerkknoten **102B** kann auch in einem Cloud-Computing-Dienstmodell betrieben werden, beispielsweise Software als Dienstleistung (SaaS), Plattform als Dienstleistung (PaaS) oder Infrastruktur als Dienstleistung (IaaS). Der Netzwerkknoten **102B** kann sich auch in einem Cloud-Computing-Einsatzmodell befinden, beispielsweise in einer privaten Cloud, einer gemeinschaftlichen Cloud, einer öffentlichen Cloud oder einer kombinierten Cloud.

[0030] Bei dem Software-Programm **108A**, **B** kann es sich um ein beliebiges von mehreren Software-Programmen höherer Ebene (higher-level software programs) handeln, die aus den durch das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten eingerichteten Verbindungen Nutzen ziehen.

[0031] Gemäß der vorliegenden Ausführungsform kann es sich bei dem Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten um ein Programm handeln, das eine Rendezvous-Funktion ausführen kann, um die Identität eines entfernt angeordneten Host-Knotens unter Verwenden eines Multicast-Protokoll zu erkennen und anschließend Sätze von Warteschlangenpaaren zur Datenkanalübertragung zu verbinden. Das Verfahren zum Erkennen und Zuteilen von Knoten wird unter Bezugnahme auf **Fig. 2** ausführlicher erläutert. In einer Ausführungsform wird das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten in der Gerätetreiberschicht des Betriebssystems für die Datenverarbeitungsknoten **102 A**, **B** ausgeführt.

[0032] Unter Bezugnahme auf **Fig. 2** wird nun ein Funktionsablaufplan gezeigt, der ein Verfahren zum Erkennen entfernt angeordneter Knoten und zum Initialisieren von Datenkanälen veranschaulicht. Bei **202** bereitet das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten die Hardware zum Verbinden von Punkt-zu-Punkt-Verbindungen zwischen Systemen vor. Dieser Adapterinitialisierungsschritt kann in drei Phasen unterteilt werden; Phase **1** kann beinhalten, dass das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten ein Zurücksetzen der Funktionsebene mit der durch den Adapter verwendeten virtuellen Funktion auslöst, um den Adapter in einen bekannten Zustand zu versetzen. Wenn für den Adapter keine solchen Kanäle definiert sind, kann das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten das Initialisieren abbrechen. Als Nächstens kann das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten den Adapter zu DMA-Transaktionen sowie MSI-X-Unterbrechungen berechtigen.

[0033] Phase **2** der Adapterinitialisierung kann beinhalten, dass das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten den Übertragungskanal einrichtet, mit Hilfe dessen Befehle an den und vom Adapter weitergeleitet werden. Phase **3** der Adapterinitialisierung kann beinhalten, dass das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten den Adapter nach entscheidenden Hardware-Konfigurationsdaten anfragt, die zu einem späteren Zeitpunkt in Befehlen an den Adapter verwendet werden können. Des Weiteren registriert das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten eine Ereigniswarteschlange und einen Speicherschutz-Tabelleneintrag, um über Ereignisse mit dem Adapter benachrichtigt zu werden und ferne Direktzugriffsspeicher- (RDMA-) Transaktionen zu ermöglichen. Das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten reserviert zu diesem Zeitpunkt außerdem QPs in Kontingenten, die sodann in Sätze unterteilt werden, wobei ein Datenkanal aus einem Satz von QPs besteht. Das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten ordnet die Sätze von QPs sequenziell zu, um das Initialisieren zu vereinfachen und die Datenflussschnittstellen abzubauen. Sobald der Adapter für Benachrichtigungen, DMAs, Unterbrechungen und anderen derartigen Datenverkehr vorbereitet worden ist, kann das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten die Datenflussschnittstellen einrichten, über die Erkennungs-, Zeitsteuerungs- und Verwaltungsvorgänge laufen sowie der Datenverkehr fließen kann.

[0034] In einer Ausführungsform der Erfindung kann das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten während Phase **3** des Adapter-Initialisierungsprozesses im Vorfeld Puffer-Ressourcen vom Adapter reservieren. Das Reservieren von Pufferpools ist ein kostspieliger Vorgang; in dieser Ausführungsform kann der Pufferpool-Reservierungsvorgang nur einmal während der Adapterinitialisierung ausgeführt werden, wodurch die Ressourcen später zum schnellen Zuordnen verfügbar sind. In einer Ausführungsform steuert das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten die Pufferpoolreservierung unter Verwenden einer als Single-Root-E/A-Virtualisierung (single-root in/out virtualization, SRIOV) bezeichneten Schnittstelle, bei der es sich um eine PCI-Express-Erweiterung handelt, die es einer Einheit wie etwa einem Netzwerkadapter ermöglicht, unter verschiedenen PCIe-Hardware-Funktionen separat auf seine Ressourcen zuzugreifen. Unter dieser Schnittstelle kann ein Einheitentreiber der Verwaltungseigner der Ressourcen sein, und das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten kann der Gast sein.

[0035] Als Nächstes erkennt das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten bei 204 entfernt angeordnete Systeme auf der anderen Seite der Verbindung durch einen Erkennungsprozess. Um das Erkennen des Knotens/entfernt angeordneten Hosts auszuführen, kann das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten eine Multicast-MAC-Zieladresse zusammen mit proprietären Nutzdaten verwenden. Wenn ein Anschluss hinzukommt und Host-Firmware konfiguriert wurde, kann das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten eine QP einrichten, auch als Datenflussschnittstelle bekannt, und in regelmäßigen Abständen eine Nachricht mit der Multicast-MAC-Adresse und Nutzdaten in Form eines unbearbeiteten Ethernet-Pakets versenden. Neben dem Einrichten der Datenflussschnittstelle kann das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten die MAC-Adresse in der Hardware konfigurieren, um zu erkennen, wann der Datenverkehr mit der angegebenen MAC-Adresse weitergeleitet werden muss. Sobald die ferne Host-Firmware bereit ist und ausgeführt wird, kann der entfernt angeordnete Host das Ethernet-Paket mit dieser angegebenen MAC-Adresse empfangen und die Nutzda-

ten speichern. Die Nutzdaten enthalten Informationen über die entfernt angeordnete Seite der Verbindung und können zum Überprüfen der Verbindung verwendet werden. Um das Übertragen von Netzwerkdatenverkehr zu ermöglichen, kann das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten zusätzlich zu Verbindungsinformationen Informationen übertragen, die verwendet werden, um zu entscheiden, welche Seite der Verbindung mit den übergeordneten Initialisierungsprozeduren beginnt. In einer Ausführungsform können diese Zuteilungsinformationen aus Zeitstempeldaten bestehen. Neben den Zuteilungsinformationen können die über die Nutzdaten übertragenen Informationen enthalten: eine Ethernet-Adresse, einen globalen Kennzeichner, eine Verwaltungs-QP-Nummer, ein Sysplex Time Protocol- (STP-) QP, RDMA-Schlüsselinformationen und eine Host-Firmware-Version. Die Ethernet-Adresse und der Gruppenkennzeichner (group identifier, GID) können die kennzeichnenden Informationen zum Übertragen eines gerichteten Netzwerkdatenverkehrs sowohl Punkt-zu-Punkt als auch in einer vernetzten Vermittlungsumgebung sein. Das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten kann die QP-Nummern und die RDMA-Schlüsselinformationen verwenden, um zu ermitteln, wohin das Paket übertragen werden muss, sobald es an einem bestimmten Endpunktadapter angekommen ist. Schließlich kann die Firmware-Version im Steuerblock eingestellt werden, und das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten kann die Firmware-Version zum Identifizieren einer unterstützten Codefunktion verwenden.

[0036] Bei **206** erzeugt das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten anschließend Verwaltungs-Datenverkehrswege. Bei den Verwaltungs-Datenverkehrswegen kann es sich um Übertragungskanäle für Netzwerkdatenverkehr handeln, die allgemeine Pfade für erforderliche spezifische Protokolle bereitstellen. Dazu können Daten erfassende Protokolle wie etwa ein Zeitsynchronisierungsprotokoll oder ein Protokoll zum Sysplex-weiten Protokollieren gehören. In einer alternativen Ausführungsform können die Verwaltungsspuren mehr als ein Startprotokoll, beispielsweise STP, verwenden, die nicht kanalspezifisch sind.

[0037] Schritte **208** bis **216** weisen die Kanalinitialisierungsprozedur auf. Sobald alle Schnittstellen auf Anschlussebene/Datenflussschnittstellen in vorangegangenen Schritten eingerichtet wurden, ist der nächste Schritt die Kanalinitialisierungsprozedur. Die Einheit, die diese Kanalinitialisierungsprozedur ausführt, ist als Kanalinitialisierungs-Zuteilungseinrichtung (channel initialization arbiter) bekannt und kann ermitteln, welche Kanäle gültig sind, sich nicht in einem Fehlerzustand befinden oder mit einem Schwellenwert versehen sind und sich im richtigen Architekturzustand befinden. Für jeden definierten Kanal startet die Kanalinitialisierungs-Zuteilungseinrichtung einen Zeitgeber und beginnt mit der Kanalinitialisierungsprozedur, die die Schritte aufweist: Prüfen der Verbindung, Verbinden des zusätzlichen Warteschlagenpaars, Verbinden der Puffer und Austauschen der Knotendeskriptoren.

[0038] Bei **208** prüft das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten die Verbindung zwischen den Knoten. Die Phase des Prüfens der Verbindung dient dazu sicherzustellen, dass der Knoten, den das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten erkannt hat, korrekt ist. Die Phase kann zunächst die Aktion der aktiven Seite der Verbindung beinhalten, die das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten unter Verwenden der Zuteilungsinformationen im Knotenerkennungsabschnitt der Verbindungsinitialisierung eingerichtet hat; als Erstes überträgt das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten die Definition des Kanals des lokalen Knotens (aktive Seite) an den entfernt angeordneten Knoten (passive Seite). Die Kanaldefinition kann aus den beiden Kennzeichnern bestehen, die als Kanalteilsystemkennung (channel sub-system identification, CSSID) und Kanalnummer (channel path identifier, CHPID) bekannt sind. Auf der passiven Seite werden die Kanalinformationen der aktiven Seite und die Kanalinformationen der passiven Seite auf Übereinstimmung geprüft. Wenn die Informationen übereinstimmen, kann eine positive Antwort gegeben werden, und die Informationen der passiven Seite können an die aktive Seite übertragen werden. Wenn irgendwelche der Informationen nicht übereinstimmen, kann ein Antwortcode an die aktive Seite rückübertagen werden, um den Fehler anzuzeigen, und ein spezieller Fehlercode kann an den Benutzer übertragen werden. Die Pakete zum Überprüfen der Verbindung werden unter Verwenden des Verwaltungs-QP übertragen.

[0039] Anschließend initiiert das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten bei 210 die zusätzliche Datenflussschnittstelle (AUX QP) des Kanals. Diese Datenflussschnittstelle wird für kanalspezifische Nichtdaten-Informationen (non-data information) verwendet. Zu kanalspezifischen Informationen gehören Überwachungssignale und geringfügige Wiederherstellungsmaßnahmen zum Verwalten dieses Kanals, ohne andere Kanäle am Anschluss zu beeinflussen. Das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten richtet die AUX QP-Datenflussschnittstelle unter Verwenden der gleichen Verbindungsprozedur ein, die im nächsten Schritt beschrieben wird. Neben dem Einrichten der AUX QP-Datenflussschnittstelle wählt das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten einen Satz von QPs aus, die zuvor vom Adapter reserviert wurden, und ordnet entsprechende Steuerspeicherbereiche zu. Außerdem richtet das Programm **110A**,

110B zum Erkennen und Zuteilen von Knoten eine Durchführungswarteschlange (completion queue) ein, die verwendet wird, um die Host-Firmware zu benachrichtigen, dass ein neues Paket in einer der dem Kanal zugeordneten QPs angekommen ist und verarbeitet werden muss.

[0040] Als Nächstes handelt das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten bei 212 den Puffer aus und verbindet diesen. In einer Ausführungsform verwendet das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten einen vierstufigen Prozess, der folgendermaßen beschrieben wird: Zunächst gibt die Host-Firmware sowohl der aktiven als auch der passive Seite jede Art von QP an, das jeweils zum Verbinden benötigt wird. Mit diesen Informationen verbindet die Firmware jeder Seite sodann der Reihe nach jedes QP. Zum Aushandeln der Anzahl zu verwendender QPs wird unter Verwenden des Verwaltungs-QP eine Nachricht von der aktiven Seite an die passive Seite übertragen, die die maximale Anzahl von QPs enthält, die sie unterstützt. Die passive Seite vergleicht sodann die Werte mit den ihr vorliegenden, und wenn diese Werte die gleichen sind, rücküberträgt die passive Seite die Werte an die aktive Seite. Wenn die Werte nicht die gleichen sind, wird der geringere der beiden Werte an die aktive Seite rückübertragen. Die aktive Seite prüft sodann die empfangene QP-Anzahl und geht anschließend die ausgehandelte Anzahl von QPs der Reihe nach durch, wobei jedes QP pro Seite in zwei Schritten eingerichtet wird. Das QP auf der aktiven Seite wird aus dem Rücksetzzustand eingerichtet und initialisiert, und anschließend überträgt die aktive Seite eine Verbindungsanforderung über das Verwaltungs-QP an die passive Seite, wobei das entsprechende QP aus dem Rücksetzzustand initialisiert wird. Sobald die passive Seite abgeschlossen ist, rücküberträgt die passive Seite eine Verbindungsantwort an die aktive Seite, und das aktive QP wird zum Empfangen von Datenverkehr vorbereitet, anschließend zum Übertragen und erneuten Übertragen einer Verbindungsbestätigung an die passive Seite vorbereitet. Die passive Seite führt sodann die beiden gleichen Schritte für ihr QP aus und rücküberträgt eine ‚Verbindung erfolgt'-Nachricht an die aktive Seite, wobei die aktive Seite anschließend die Verbindung des nächsten QP initialisiert.

[0041] Bei **214** tauscht das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten sodann Knotendeskriptoren über das Verwaltungs-QP aus. Dadurch werden der anderen Seite spezifische Kanalinformationen zur Verfügung gestellt, und der Beginn des Datenverkehrs wird ermöglicht. Bei den Informationen handelt es sich um eine einfache Ping-Pong-Nachricht von der aktiven Seite an die passive Seite und zurück. Sobald dieser Schritt erfolgt ist, ist der Kanal initialisiert, und das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten führt einen Test durch, um sicherzustellen, dass Datenverkehr übertragen werden kann und der Kanal nominell und ohne Fehler funktioniert. Das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten kann das Verbindungsinitialisierungsprogramm am Back-End aufrufen, um das Softwareprogramm **108A**, B über die Verfügbarkeit des Kanals zu benachrichtigen.

[0042] Bei **216** ermittelt das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten sodann, ob alle der definierten Kanäle unter Verwenden der Kanalinitialisierungs-Zuteilungsfunktion initialisiert wurden. Das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten ermittelt dies, indem es die Liste definierter Kanäle durchgeht. Gemäß einer Ausführung kann das Verfahren enden, wenn die Kanalinitialisierungs-Zuteilungsfunktion feststellt, dass alle Kanäle initialisiert wurden (Schritt **216**, „JA"-Verzweigung). Wenn die Kanalinitialisierungs-Zuteilungsfunktion feststellt, dass nicht alle Kanäle initialisiert wurden (Schritt **216**, „NEIN"-Verzweigung), kann das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten zum Schritt **208** weitergehen, um die Kanalinitialisierungsprozedur erneut zu starten und die Verbindung zwischen den Knoten zu überprüfen.

[0043] **Fig. 3** ist ein funktionelles Blockschaubild, das eine Verbindungsinitialisierungsprozedur gemäß mindestens einer Ausführungsform veranschaulicht. Die Leistungsfähigkeit einer Adapterinitialisierung **304** wird im Schritt **202** von **Fig. 2** ausführlicher beschrieben. Während der Adapterinitialisierung **304** wird in einer Ausführungsform der Erfindung eine Anschlussinitialisierung **306** in den Schritten **204** und **206** von **Fig. 2** ausführlicher beschrieben. Eine Kanalinitialisierung **308** wird in den Schritten **208** bis **216** von **Fig. 2** näher erläutert. Während der Kanalinitialisierung **308** wird die Kanalinitialisierungs-Zuteilungseinrichtung verwendet, um zu ermitteln, welche Kanäle gültig sind, sich nicht in einem Fehlerzustand befinden oder mit einem Schwellenwert versehen sind und sich im richtigen Architekturzustand befinden. Nach der Kanalinitialisierung **308** übergibt der Prozess den neu erzeugten virtuellen Kanal an das Software-Programm **108**.

[0044] Es sollte klar sein, dass die **Fig. 2** bis **Fig. 3** lediglich eine Veranschaulichung einer Ausführung zeigen und keinerlei Beschränkungen in Bezug darauf bedeuten, wie verschiedene Ausführungsformen realisiert werden können. An den gezeigten Umgebungen können je nach Anforderungen an Gestaltung und Ausführung viele Modifikationen vorgenommen werden.

[0045] **Fig. 4** ist ein Blockschaltbild **400** von internen und externen Komponenten des in **Fig. 1** abgebildeten Netzwerkknotens **102A**, **102B** gemäß einer Ausführungsform der vorliegenden Erfindung. Es sollte

klar sein, das **Fig. 4** lediglich eine Veranschaulichung einer Ausführung zeigt und keinerlei Beschränkungen in Bezug auf die Umgebungen bedeutet, in denen verschiedene Ausführungsformen realisiert werden können. An den gezeigten Umgebungen können je nach Anforderungen an Gestaltung und Ausführung viele Modifikationen vorgenommen werden.

[0046] Das Datenverarbeitungssystem **402**, **404** ist repräsentativ für jede elektronische Einheit, die in der Lage ist, durch Maschinen lesbare Programmanweisungen auszuführen. Das Datenverarbeitungssystem **402**, **404** kann repräsentativ für ein Smartphone, ein Computersystem, einen PDA oder andere elektronische Einheiten sein. Zu Beispielen für Datenverarbeitungssysteme, -umgebungen und/oder -konfigurationen, die durch das Datenverarbeitungssystem **402**, **404** repräsentiert werden können, gehören Personal-Computer-Systeme, Server-Computersysteme, Thin Clients, Thick Clients, Handheld- oder Laptop-Computer, Multiprozessorsysteme, auf Mikroprozessoren beruhende Systeme, Netzwerk-PCs, Minicomputer-Systeme und verteilte Cloud-Computing-Umgebungen, die eine der obigen Systeme oder Einheiten enthalten, jedoch ohne auf diese beschränkt zu sein.

[0047] Der Netzwerkknoten **102A**, **102B** kann entsprechende Sätze interner Komponenten **402** a,b und externer Komponenten **404** a,b enthalten, die in **Fig. 4** gezeigt werden. Jeder der Sätze interner Komponenten **402** enthält einen oder mehrere Prozessoren **420**, einen oder mehrere durch einen Computer lesbare RAMs **422** und einen oder mehrere durch einen Computer lesbare ROMs **424** auf einem oder mehreren Bussen **426** und ein oder mehrere Betriebssysteme **428** und eine oder mehrere durch einen Computer lesbare physische Speichereinheiten **430**. Das eine oder die mehreren Betriebssysteme **428**, das Software-Programm **108** und das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten im Netzwerkknoten **102A**, **102B** sind in einer oder mehreren der jeweiligen durch einen Computer lesbaren physischen Speichereinheiten **530** zum Ausführen durch einen oder mehrere der jeweiligen Prozessoren **420** über einen oder mehrere der jeweiligen RAMs **422** (die üblicherweise einen Cachespeicher enthalten) gespeichert. In der in **Fig. 4** gezeigten Ausführungsform handelt es sich bei jeder der durch einen Computer lesbaren physischen Speichereinheiten **430** um eine magnetische Plattenspeichereinheit eines internen Festplattenlaufwerks. Alternativ handelt es sich bei jeder der durch einen Computer lesbaren physischen Speichereinheiten **430** um eine Halbleiterspeichereinheit wie etwa den ROM **424**, einen EPROM, einen Flash-Speicher oder eine andere durch einen Computer lesbare physische Speichereinheit, die ein Computerprogramm und digitale Daten speichern kann.

[0048] Jeder Satz von internen Komponenten **402** a,b enthält außerdem ein Lese/Schreib-Laufwerk (R/W drive) oder eine Schnittstelle **432**, um aus einer oder mehreren tragbaren, durch einen Computer lesbaren physischen Speichereinheiten **438** wie etwa einer CD-ROM, einer DVD, einem Speicherstick, einem Magnetband, einer Magnetplatte, einer optischen Platte oder einer Halbleiterspeichereinheit zu lesen oder in diese zu schreiben. Ein Software-Programm, beispielsweise das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten, kann in einer oder mehreren der jeweiligen tragbaren, durch einen Computer lesbaren physischen Speichereinheiten **438** gespeichert, über das entsprechende Lese/Schreib-Laufwerk oder die Schnittstelle **432** gelesen und in das jeweilige Festplattenlaufwerk **430** geladen werden.

[0049] Jeder Satz von internen Komponenten **402** a,b enthält außerdem Netzwerkadapter oder Schnittstellen **436** wie etwa TCP/IP-Adapterkarten, drahtlose WLAN-Schnittstellenkarten oder drahtlose 3G- oder 4G-Schnittstellenkarten oder andere leitungsgebundene oder drahtlose Übertragungsverbindungen. Das Software-Programm **108** und das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten im Netzwerkknoten **102A**, **102B** kann von einem externen Computer über ein Netzwerk (zum Beispiel das Internet, ein lokales Netz oder ein Weitverkehrsnetz) und entsprechende Netzwerkadapter oder Schnittstellen **536** in den Netzwerkknoten **102A**, **102B** heruntergeladen werden. Von den Netzwerkadaptern oder Schnittstellen **436** werden das Software-Programm **108** und das Programm **110A**, **110B** zum Erkennen und Zuteilen von Knoten im Netzwerkknoten in das jeweilige Festplattenlaufwerk **430** geladen. Das Netzwerk kann Kupferleitungen, Lichtwellenleiter, drahtlose Übertragung, Leitwegrechner, Firewalls, Vermittlungseinheiten, Gateway-Computer und/oder Edge-Server aufweisen.

[0050] Jeder der Sätze von externen Komponenten **404** a,b kann einen Computerbildschirm **444**, eine Tastatur **442** und eine Computermaus **434** enthalten. Die externen Komponenten **404** a,b können außerdem berührungsempfindliche Bildschirme, virtuelle Tastaturen, Touchpads, Zeigereinheiten und andere Einheiten mit direkter Benutzerinteraktion (human interface devices) enthalten. Jeder der Sätze von internen Komponenten **402** a,b enthält außerdem Einheitentreiber **440** zum Anschließen an den Computerbildschirm **444**, die Tastatur **442** und die Computermaus **434**. Die Einheitentreiber **440**, das Lese/Schreib-Laufwerk oder die Schnittstelle **432** und der Netzwerkadapter oder die Schnittstelle **436** weisen Hardware und Software auf (gespeichert in der Speichereinheit **430** und/oder im ROM **424**).

[0051] Zwar beinhaltet diese Beschreibung eine ausführliche Beschreibung von Cloud Computing,

es sollte jedoch von vornherein klar sein, dass Implementierungen der hierin wiedergegebenen Lehre nicht auf eine Cloud-Computing-Umgebung beschränkt sind. Vielmehr können Ausführungsformen der vorliegenden Erfindung in Verbindung mit einem beliebigen anderen Typ von gegenwärtig bekannter oder zu einem späteren Zeitpunkt entwickelter Datenverarbeitungsumgebung implementiert werden.

[0052] Cloud-Computing ist ein Dienstleistungsmodell, das einen komfortablen und bedarfsorientierten Netzwerkzugang zu einem gemeinsam genutzten Pool von konfigurierbaren Datenverarbeitungs-Ressourcen ermöglicht (z.B. Netzwerke, Netzwerk-Bandbreite, Server, Verarbeitung, Speicher, Speicherplatz, Anwendungen, virtuelle Maschinen und Dienste), die mit minimalem Verwaltungsaufwand bzw. minimaler Interaktion mit einem Anbieter des Dienstes schnell zur Verfügung gestellt und freigegeben werden können. Dieses Cloud-Modell kann mindestens fünf Charakteristika, mindestens drei Dienstmodelle und mindestens vier Bereitstellungsmodelle beinhalten.

[0053] Bei den Charakteristika handelt es sich um die folgenden:

[0054] Bedarfsorientierte Selbstbedienung (on-demand self-service): Ein Cloud-Kunde kann je nach Bedarf einseitig Datenverarbeitungs-Ressourcen wie Server-Zeit und Netzspeicher automatisch in Anspruch nehmen, ohne dass eine menschengeführte Interaktion mit dem Anbieter des Dienstes erforderlich ist.

[0055] Allgemeiner Netzzugriff (broad network access): Ressourcen sind über ein Netzwerk verfügbar und über Standardmechanismen zugänglich, die eine Verwendung durch heterogene Thin- oder Thick-Client-Plattformen (z.B. Mobiltelefone, Laptops und PDAs) unterstützen.

[0056] Ressourcen-Bündelung (resource Pooling): Die Datenverarbeitungs-Ressourcen des Anbieters werden gebündelt, um mehreren Kunden zu dienen, wobei ein Multi-Tenant-Modells verwendet wird, bei dem verschiedene physische und virtuelle Ressourcen entsprechend dem Bedarf dynamisch zugewiesen und erneut zugewiesen werden. Es besteht insofern eine gewisse Unabhängigkeit vom Standort, als der Kunde im Allgemeinen keine Kontrolle über den bzw. Kenntnis vom genauen Standort der bereitgestellten Ressourcen hat, den Standort jedoch auf einer höheren Abstraktionsebene (z.B. Region, Staat oder Datenzentrum) festlegen kann.

[0057] Schnelle Anpassungsfähigkeit (rapid elasticity): Ressourcen können rasch und anpassungsfähig, in einigen Fällen automatisch, bereitgestellt werden, um schnell eine Erweiterungsmöglichkeit (scale out) zu bieten, und rasch wieder freigegeben werden, wodurch eine schnelle Reduzierung (scale in) geboten wird. Aus Sicht des Kunden scheinen die zur Bereitstellung verfügbaren Ressourcen oftmals unbegrenzt zu sein und können in beliebiger Menge zu einem beliebigen Zeitpunkt erworben werden.

[0058] Dienst für nutzungsabhängige Abrechnung (measured service): Cloud-Systeme steuern und optimieren die Ressourcen-Nutzung automatisch durch Einsatz einer Messfunktion auf einer Abstraktionsebene, die für den Typ des Dienstes (z.B. Speicherplatz, Verarbeitung, Bandbreite sowie aktive Benutzerkonten) geeignet ist. Die Ressourcen-Nutzung kann überwacht, gesteuert und aufgezeichnet werden, wobei sowohl für den Anbieter als auch für den Abnehmer des verwendeten Dienstes Transparenz gegeben ist.

[0059] Bei den Dienstmodellen handelt es sich um die folgenden:

[0060] Software als Dienstleistung (Software as a Service, SaaS): Die für den Kunden bereitgestellte Funktionalität besteht in der Nutzung der in einer Cloud-Infrastruktur ausgeführten Anwendungen des Anbieters. Auf die Anwendungen kann von verschiedenen Client-Einheiten durch eine Thin-Client-Schnittstelle wie einen Web-Browser (z.B. auf dem Web beruhende eMail) zugegriffen werden. Der Kunde verwaltet bzw. steuert nicht die zugrunde liegende Cloud-Infrastruktur, darunter Netzwerk, Server, Betriebssysteme, Speicherplatz oder sogar einzelne Anwendungsmöglichkeiten, mit der möglichen Ausnahme begrenzter benutzerspezifischer Anwendungskonfigurationseinstellungen.

[0061] Plattform als Dienstleistung (Platform as a Service, PaaS): Die für den Kunden bereitgestellte Funktionalität besteht in der Verwendung von vom Kunden erstellten oder angeforderten Anwendungen, die unter Verwenden von vom Anbieter unterstützten Programmiersprachen und Hilfsprogrammen erzeugt werden, in der Cloud-Infrastruktur. Der Kunde verwaltet oder steuert nicht die zugrundeliegende Cloud-Infrastruktur, darunter Netze, Server, Betriebssysteme oder Speicherplatz, hat jedoch die Kontrolle über die eingesetzten Anwendungen und möglicherweise über Konfigurationen der Hosting-Umgebung der Anwendung.

[0062] Infrastruktur als Dienstleistung (Infrastructure as a Service, IaaS): Die für den Kunden bereitgestellte Funktionalität besteht in der Bereitstellung von Datenverarbeitung, Speicherplatz, Netzwerken und anderen grundlegenden Datenverarbeitungs-Ressourcen, wobei der Kunde eine beliebige Software einsetzen und ausführen kann, die Betriebssysteme und Anwendungen beinhalten kann. Der Kunde verwaltet oder steuert nicht die zugrunde liegende Cloud-

Infrastruktur, hat jedoch die Kontrolle über Betriebssysteme, Speicherplatz, eingesetzte Anwendungen und möglicherweise eine begrenzte Kontrolle über die Auswahl von Netzwerkkomponenten (z.B. Host-Firewalls).

[0063] Bei den Einsatzmodellen handelt es sich um die folgenden:

[0064] Private Cloud: Die Cloud-Infrastruktur wird einzig und allein für eine Organisation betrieben. Sie kann durch die Organisation selbst oder einen Dritten verwaltet werden und kann sich in den eigenen Räumen oder in fremden Räumen befinden.

[0065] Gemeinschaftliche Cloud (community cloud): Die Cloud-Infrastruktur wird von mehreren Organisationen gemeinsam genutzt und unterstützt eine spezielle Benutzergemeinschaft mit gemeinsamen Interessen (z.B. Zielsetzung, Sicherheitsanforderungen, Strategie- und Konformitätsüberlegungen). Sie kann durch die Organisationen oder einen Dritten verwaltet werden und kann sich in den eigenen Räumen oder fremden Räumen befinden.

[0066] Öffentliche Cloud (public cloud): Die Cloud-Infrastruktur wird der Allgemeinheit oder einer großen Industriegruppe zur Verfügung gestellt und ist im Besitz einer Organisation, die Cloud-Dienste verkauft.

[0067] Kombinierte Cloud (hybrid cloud): Die Cloud-Infrastruktur setzt sich aus zwei oder mehreren Clouds (privat, gemeinschaftlich oder öffentlich) zusammen die eigenständige Einheiten bleiben, jedoch durch eine standardisierte oder firmeneigene Technologie, die die Übertragbarkeit von Daten und Anwendungen ermöglicht, miteinander verbunden sind (z.B. Cloud-Zielgruppenverteilung für eine Auslastungsverteilung zwischen Clouds).

[0068] Eine Cloud-Computing-Umgebung ist dienstleistungsorientiert mit Fokus auf Statusunabhängigkeit, loser Kopplung, Modularität und semantischer Interoperabilität. Den Kern des Cloud-Computing bildet eine Infrastruktur, die ein Netzwerk aus miteinander verbundenen Knoten aufweist.

[0069] Unter Bezugnahme auf **Fig. 5** ist nun die veranschaulichende Cloud-Computing-Umgebung **60** abgebildet. Wie gezeigt ist, weist die Cloud-Computing-Umgebung **50** einen oder mehrere Cloud-Computing-Knoten **100** auf, mit denen von Cloud-Kunden verwendete lokale Datenverarbeitungseinheiten wie der Personal Digital Assistant (PDA) oder das Mobiltelefon **54A**, der Tisch-Computer **54B**, der Laptop-Computer **54C** und/oder das Automobil-Computer-System **54N** Daten austauschen können. Die Knoten **100** können miteinander Daten austauschen. Sie können physisch oder virtuell in ein oder mehrere Netzwerke wie private, gemeinschaftliche, öffentliche oder kombinierte Clouds gruppiert werden (nicht gezeigt), wie vorstehend beschrieben wurde, oder in eine Kombination daraus. Dies ermöglicht es der Cloud-Computing-Umgebung **50**, Infrastruktur, Plattformen und/oder Software als Dienstleistungen anzubieten, für die ein Cloud-Kunde keine Ressourcen auf einer lokalen Datenverarbeitungseinheit vorhalten muss. Es sei klargestellt, dass die Arten von in **Fig. 5** gezeigten Datenverarbeitungseinheiten **54A** bis N lediglich veranschaulichend sein sollen und dass die Datenverarbeitungsknoten **100** und die Cloud-Computing-Umgebung **50** über eine beliebige Art Netzwerk und/oder über eine beliebige Art von über ein Netzwerk aufrufbarer Verbindung (z.B. unter Verwenden eines Web-Browsers) mit einer beliebigen Art von computergestützter Einheit Daten austauschen können.

[0070] Unter Bezugnahme auf **Fig. 6** wird nun ein Satz von funktionalen Abstraktionsschichten **600** gezeigt, die durch die Cloud-Computing-Umgebung **50** bereitgestellt werden. Es sollte von vornherein klar sein, dass die in **Fig. 6** gezeigten Komponenten, Schichten und Funktionen lediglich veranschaulichend sein sollen und Ausführungsformen der Erfindung nicht darauf beschränkt sind. Wie abgebildet ist, werden die folgenden Schichten und entsprechenden Funktionen bereitgestellt:

[0071] Eine Hardware- und Software-Schicht **60** enthält Hardware- und Software-Komponenten. Zu Beispielen für Hardware-Komponenten gehören: Mainframe-Computer **61**; auf der RISC-(Reduced Instruction Set Computer) Architektur beruhende Server **62**; Server **63**; Blade-Server **64**; Speichereinheiten **65**; und Netzwerke sowie Netzwerkkomponenten **66**. In einigen Ausführungsformen beinhalten Software-Komponenten eine Netzwerk-Anwendungsserver-Software **67** und eine Datenbank-Software **68**.

[0072] Die Virtualisierungsschicht **70** stellt eine Abstraktionsschicht bereit, aus der die folgenden Beispiele für virtuelle Einheiten bereitgestellt werden können: virtuelle Server **71**, virtueller Speicher **72**, virtuelle Netzwerke **73**, darunter virtuelle private Netzwerke, virtuelle Anwendungen und Betriebssysteme **74**; und virtuelle Clients **75**.

[0073] In einem Beispiel kann die Verwaltungsschicht **80** die nachfolgend beschriebenen Funktionen bereitstellen. Die Ressourcen-Versorgung (resource provisioning) **81** stellt ein dynamisches Beschaffung von Datenverarbeitungs-Ressourcen sowie anderen Ressourcen bereit, die zum Durchführen von Aufgaben innerhalb der Cloud-Computing-Umgebung verwendet werden. Die Gebührenerfassung und Preisermittlung (metering and pricing) **82** stellt eine Kostenverfolgung beim Verwenden von Ressourcen innerhalb der Cloud-Computing-Umgebung sowie eine Abrechnung oder Rechnungsstel-

lung für die Inanspruchnahme dieser Ressourcen bereit. In einem Beispiel können diese Ressourcen Anwendungssoftware-Lizenzen aufweisen. Die Sicherheit (security) stellt eine Identitätsüberprüfung für Cloud-Kunden und Aufgaben sowie Schutz für Daten und andere Ressourcen bereit. Das Benutzerportal **83** stellt den Zugang zu der Cloud-Computing-Umgebung für Nutzer und für Systemadministratoren bereit. Die Dienstgüteverwaltung (service level management) **84** stellt eine Zuordnung und Verwaltung von Cloud-Computing-Ressourcen bereit, so dass die Anforderungen an die Dienstgüte erfüllt werden. Die Planung und Erfüllung der Dienstgütevereinbarung (SLA, Service Level Agreement planning and fulfillment) **85** stellt die Vorbereitung für und die Beschaffung von Cloud-Computing-Ressourcen bereit, für die gemäß einer SLA ein künftiger Bedarf vorausgesehen wird.

[0074] Eine Arbeitslastenschicht (workloads layer) **90** stellt Beispiele einer Funktionalität bereit, für welche die Cloud-Computing-Umgebung verwendet werden kann. Zu Beispielen für Arbeitslasten und Funktionen, die von dieser Schicht bereitgestellt werden können, gehören: Abbildung und Navigation **91**; Software-Entwicklung und Lebenszyklusverwaltung **92**; Bildungsangebot Virtuelles Klassenzimmer **93**; Datenanalyseverarbeitung **94**; Transaktionsverarbeitung **95**; und das Verfahren **96** zum Erkennen und Zuteilen von Knoten. Das Verfahren **96** zum Erkennen und Zuteilen von Knoten kann Verwenden von MAC-Multicast-Prozeduren, um eine Adressierbarkeit entfernt angeordneter Knoten zu erkennen, und von InfiniBand®-Protokollen betreffen (InfiniBand® und alle auf InfiniBand® beruhenden Warenzeichen und Logos sind Warenzeichen oder eingetragene Warenzeichen von System I/O, Incorporated und/oder seinen angegliederten Unternehmen), um beim Implementieren von einem oder mehreren Übertragungskanälen Sätze von Warteschlangenpaaren zu verbinden.

[0075] Die Beschreibungen der verschiedenen Ausführungsformen der vorliegenden Erfindung wurden zur Veranschaulichung dargelegt, sollen jedoch nicht erschöpfend oder auf die offenbarten Ausführungsformen beschränkt sein. Für Fachleute sind viele Modifikationen und Änderungen ersichtlich, ohne vom Schutzumfang der beschriebenen Ausführungsformen abzuweichen. Die hierin verwendete Terminologie wurde gewählt, um den Grundgedanken der Ausführungsform, die praktische Anwendung oder technische Verbesserung gegenüber marktüblichen Technologien bestmöglich zu erläutern oder anderen Fachleuten ein Verständnis der hierin offenbarten Ausführungsformen zu ermöglichen.

Patentansprüche

1. Durch einen Prozessor implementiertes Verfahren zum Erkennen entfernt angeordneter Knoten und zum Verbinden einer Mehrzahl von Warteschlangenpaaren zum Implementieren von Übertragungskanälen, wobei das Verfahren aufweist:
Vorbereiten einer Mehrzahl von Hardware-Komponenten zum Verbinden von einer oder mehreren Punkt-zu-Punkt-Verbindungen zwischen einer Mehrzahl von Knoten;
Erkennen der Mehrzahl von Knoten in der einen oder den mehreren Punkt-zu-Punkt-Verbindungen durch einen Erkennungsprozess;
Erzeugen einer Mehrzahl von Verwaltungs-Datenverkehrswegen zwischen der Mehrzahl von Knoten;
Prüfen einer Mehrzahl von Verbindungen zwischen der Mehrzahl von Knoten;
Initiieren von einer oder mehreren zusätzlichen Datenflussschnittstellen in der Mehrzahl von Verbindungen;
Aushandeln und Verbinden einer Mehrzahl von Puffern für die Mehrzahl von Verbindungen; und
Austauschen einer Mehrzahl von Knotendeskriptoren zwischen der Mehrzahl von Knoten.

2. Verfahren nach Anspruch 1, wobei das Verfahren in einer Einheitentreiberschicht ausgeführt wird.

3. Verfahren nach Anspruch 1, wobei Vorbereiten einer Mehrzahl von Hardware-Komponenten zum Verbinden von einer oder mehreren Punkt-zu-Punkt-Verbindungen zwischen einer Mehrzahl von Knoten außerdem aufweist:
im Vorfeld Reservieren einer Mehrzahl von Puffer-Ressourcen aus der Mehrzahl von Ha rdware-Komponenten.

4. Verfahren nach Anspruch 3, wobei Reservieren der Mehrzahl von Puffer-Ressourcen aus der Mehrzahl von Hardware-Komponenten im Vorfeld unter Verwenden einer Single-Root-E/A-Virtualisierungsschnittstelle ausgeführt wird.

5. Verfahren nach Anspruch 1, wobei Reservieren einer Mehrzahl von Puffer-Ressourcen aus der Mehrzahl von Hardware-Komponenten einmal ausgeführt wird.

6. Verfahren nach Anspruch 1, das außerdem aufweist:
Verwenden einer Kanalinitialisierungs-Zuteilungsfunktion, um sicherzustellen, dass alle aus einer Mehrzahl von Kanälen in der Mehrzahl von Verbindungen initialisiert wurden.

7. Verfahren nach Anspruch 1, wobei Aushandeln und Verbinden der Mehrzahl von Puffern einen vierstufigen Prozess zum Verbinden von Warteschlangenpaaren enthalten.

8. Computersystem zum Erkennen entfernt angeordneter Knoten und zum Verbinden einer Mehrzahl von Warteschlangenpaaren zum Implementie-

ren von Übertragungskanälen, wobei das Computersystem aufweist:
einen oder mehrere Prozessoren, einen oder mehrere durch einen Computer lesbare Speicher, ein oder mehrere durch einen Computer lesbare physische Speichermedien und Programmanweisungen, die in mindestens einem der ein oder mehreren durch einen Computer lesbaren physischen Speichermedien gespeichert sind, um durch mindestens einen der ein oder mehreren Prozessoren über mindestens einen oder ein oder mehreren Speicher ausgeführt zu werden, wobei das Computersystem ein Verfahren ausführen kann, das aufweist:
Vorbereiten von Hardware zum Verbinden von einer oder mehreren Punkt-zu-Punkt-Verbindungen zwischen einer Mehrzahl von Knoten;
Erkennen der Mehrzahl von Knoten in der einen oder den mehreren Punkt-zu-Punkt-Verbindungen durch einen Erkennungsprozess;
Erzeugen einer Mehrzahl von Verwaltungs-Datenverkehrswegen zwischen der Mehrzahl von Knoten;
Prüfen einer Mehrzahl von Verbindungen zwischen der Mehrzahl von Knoten;
Initiieren von einer oder mehreren zusätzlichen Datenflussschnittstellen in der Mehrzahl von Verbindungen;
Aushandeln und Verbinden einer Mehrzahl von Puffern für die Mehrzahl von Verbindungen; und
Austauschen einer Mehrzahl von Knotendeskriptoren zwischen der Mehrzahl von Knoten.

9. Computersystem nach Anspruch 8, wobei das Verfahren in einer Einheitentreiberschicht ausgeführt wird.

10. Computersystem nach Anspruch 8, wobei Vorbereiten einer Mehrzahl von Hardware-Komponenten zum Verbinden von einer oder mehreren Punkt-zu-Punkt-Verbindungen zwischen einer Mehrzahl von Knoten außerdem aufweist:
im Vorfeld Reservieren einer Mehrzahl von Puffer-Ressourcen aus der Mehrzahl von Ha rdware-Komponenten.

11. Computersystem nach Anspruch 10, wobei Reservieren der Mehrzahl von Puffer-Ressourcen aus der Mehrzahl von Hardware-Komponenten im Vorfeld unter Verwenden einer Single-Root-E/A-Virtualisierungsschnittstelle ausgeführt wird.

12. Computersystem nach Anspruch 8, wobei Reservieren einer Mehrzahl von Puffer-Ressourcen aus der Mehrzahl von Hardware-Komponenten einmal ausgeführt wird.

13. Computersystem nach Anspruch 8, das außerdem aufweist:
Verwenden einer Kanalinitialisierungs-Zuteilungsfunktion, um sicherzustellen, dass alle aus einer Mehrzahl von Kanälen in der Mehrzahl von Verbindungen initialisiert wurden.

14. Computersystem nach Anspruch 8, wobei Aushandeln und Verbinden der Mehrzahl von Puffern einen vierstufigen Prozess zum Verbinden von Warteschlangenpaaren beinhalten.

15. Computerprogrammprodukt zum Erkennen entfernt angeordneter Knoten und zum Verbinden einer Mehrzahl von Warteschlangenpaaren zum Implementieren von Übertragungskanälen, wobei das Computerprogrammprodukt aufweist:
ein oder mehrere durch einen Computer lesbare physische Speichermedien und Programmanweisungen, die in mindestens einem der ein oder mehreren durch einen Computer lesbaren physischen Speichermedien gespeichert sind, wobei die Programmanweisungen durch einen Prozessor ausführbar sind, wobei die Programmanweisungen den Prozessor steuern, so dass er ein Verfahren ausführt, das aufweist:
Vorbereiten einer Mehrzahl von Hardware-Komponenten zum Verbinden von einer oder mehreren Punkt-zu-Punkt-Verbindungen zwischen einer Mehrzahl von Knoten;
Erkennen der Mehrzahl von Knoten in der einen oder den mehreren Punkt-zu-Punkt-Verbindungen durch einen Erkennungsprozess;
Erzeugen einer Mehrzahl von Verwaltungs-Datenverkehrswegen zwischen der Mehrzahl von Knoten;
Prüfen einer Mehrzahl von Verbindungen zwischen der Mehrzahl von Knoten;
Initiieren von einer oder mehreren zusätzlichen Datenflussschnittstellen in der Mehrzahl von Verbindungen;
Aushandeln und Verbinden einer Mehrzahl von Puffern für die Mehrzahl von Verbindungen; und
Austauschen einer Mehrzahl von Knotendeskriptoren zwischen der Mehrzahl von Knoten.

16. Computerprogrammprodukt nach Anspruch 15, wobei das Verfahren in einer Einheitentreiberschicht ausgeführt wird.

17. Computerprogrammprodukt nach Anspruch 15, wobei Vorbereiten einer Mehrzahl von Hardware-Komponenten zum Verbinden von einer oder mehreren Punkt-zu-Punkt-Verbindungen zwischen einer Mehrzahl von Knoten außerdem aufweist:
im Vorfeld Reservieren einer Mehrzahl von Puffer-Ressourcen aus der Mehrzahl von Hardware-Komponenten.

18. Computerprogrammprodukt nach Anspruch 17, wobei Reservieren der Mehrzahl von Puffer-Ressourcen aus der Mehrzahl von Hardware-Komponenten im Vorfeld unter Verwenden einer Single-Root-E/A-Virtualisierungsschnittstelle ausgeführt wird.

19. Computerprogrammprodukt nach Anspruch 15, wobei ein Reservieren einer Mehrzahl von Puffer-Ressourcen aus der Mehrzahl von Hardware-Komponenten einmal ausgeführt wird.

20. Computerprogrammprodukt nach Anspruch 15, das außerdem aufweist:
Verwenden einer Kanalinitialisierungs-Zuteilungsfunktion, um sicherzustellen, dass alle aus einer Mehrzahl von Kanälen in der Mehrzahl von Verbindungen initialisiert wurden.

Es folgen 6 Seiten Zeichnungen

Anhängende Zeichnungen

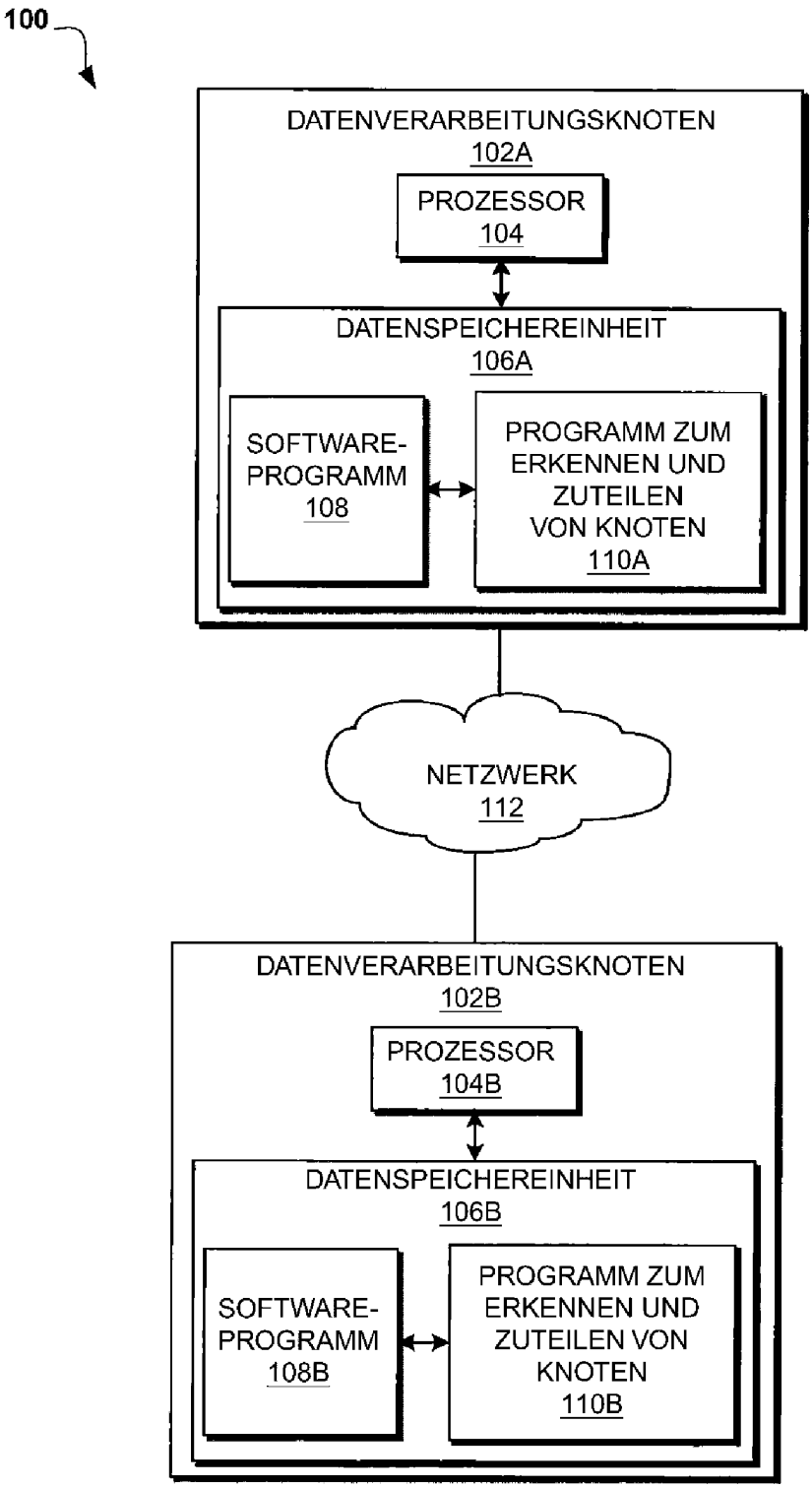


FIG. 1

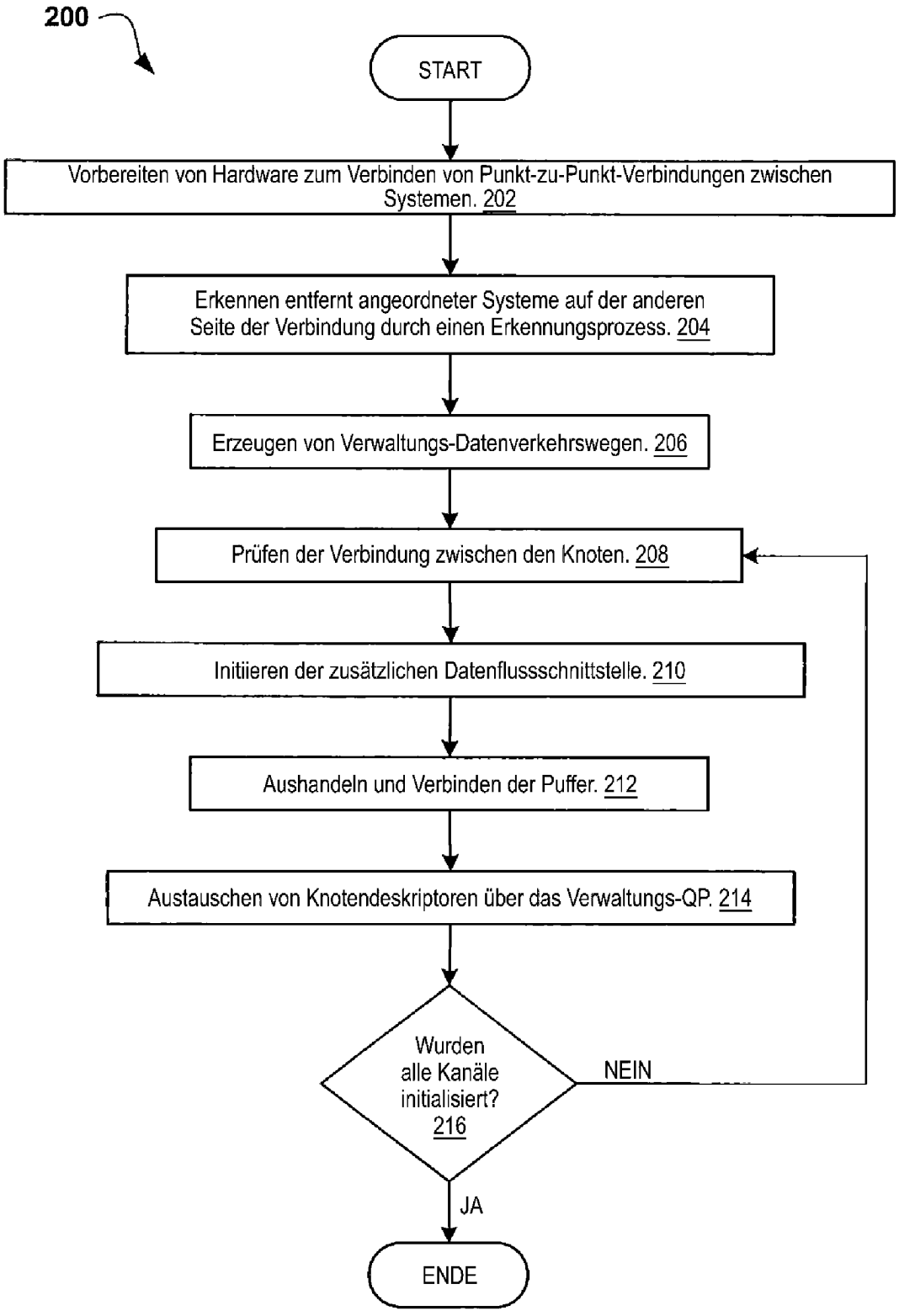

200
START
Vorbereiten von Hardware zum Verbinden von Punkt-zu-Punkt-Verbindungen zwischen Systemen. 202
Erkennen entfernt angeordneter Systeme auf der anderen Seite der Verbindung durch einen Erkennungsprozess. 204
Erzeugen von Verwaltungs-Datenverkehrswegen. 206
Prüfen der Verbindung zwischen den Knoten. 208
Initiieren der zusätzlichen Datenflussschnittstelle. 210
Aushandeln und Verbinden der Puffer. 212
Austauschen von Knotendeskriptoren über das Verwaltungs-QP. 214
Wurden alle Kanäle initialisiert? 216
NEIN
JA
ENDE

FIG. 2

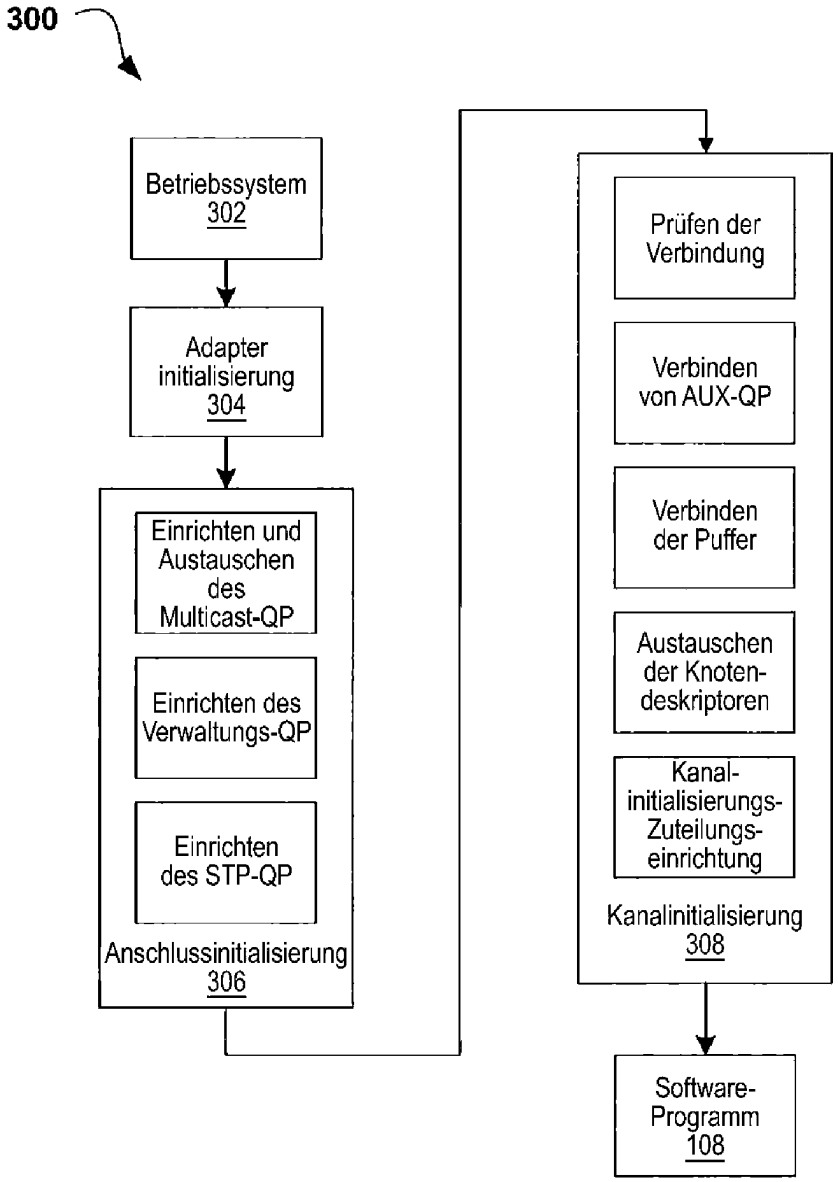

300
Betriebssystem 302
Adapter initialisierung 304
Einrichten und Austauschen des Multicast-QP
Einrichten des Verwaltungs-QP
Einrichten des STP-QP
Anschlussinitialisierung 306
Prüfen der Verbindung
Verbinden von AUX-QP
Verbinden der Puffer
Austauschen der Knoten-deskriptoren
Kanal-initialisierungs-Zuteilungs-einrichtung
Kanalinitialisierung 308
Software-Programm 108

FIG. 3

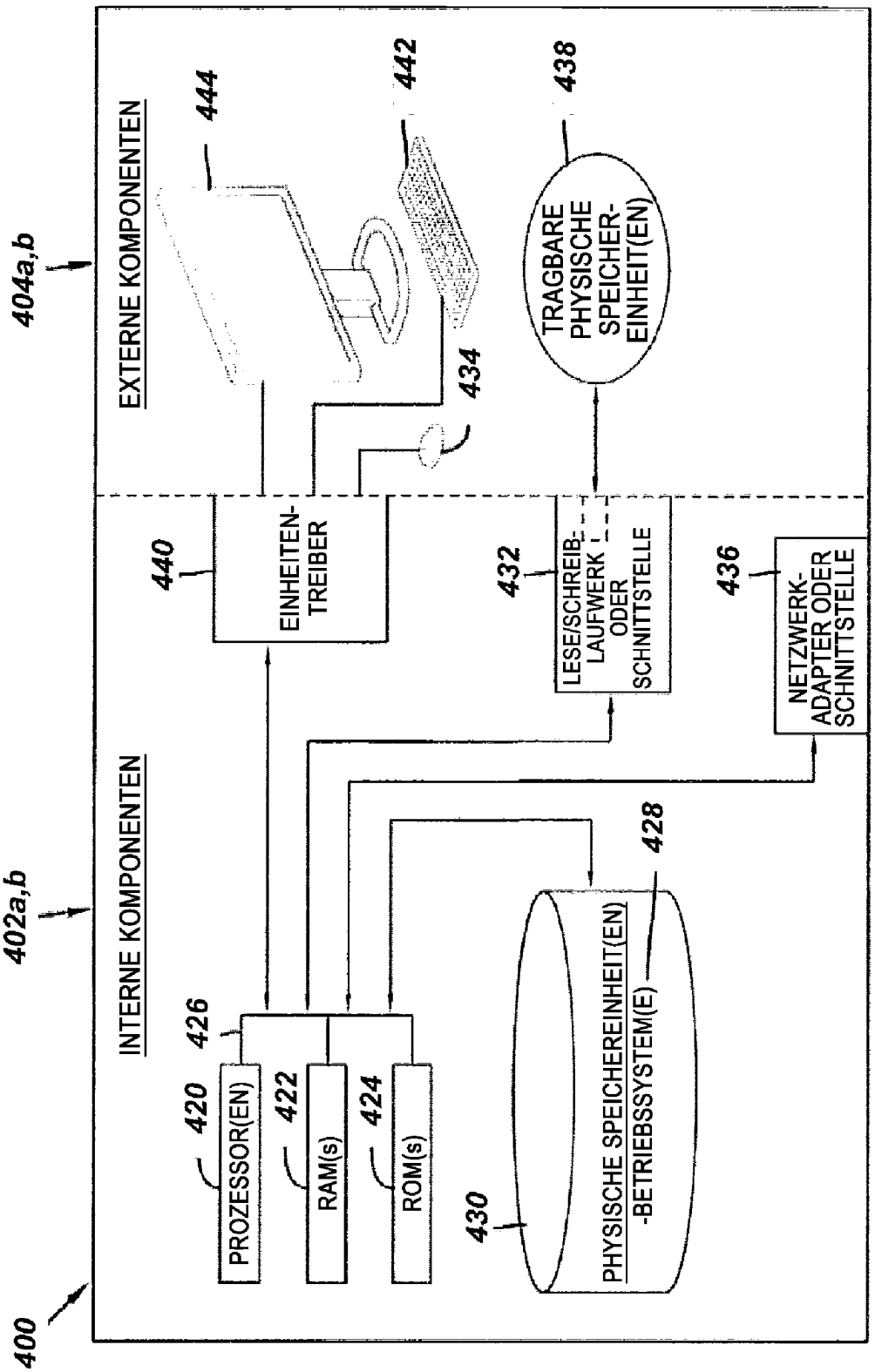

400
402a,b
404a,b
INTERNE KOMPONENTEN
EXTERNE KOMPONENTEN
PROZESSOR(EN)
RAM(s)
ROM(s)
420
422
424
426
PHYSISCHE SPEICHEREINHEIT(EN)
-BETRIEBSSYSTEM(E)
428
430
EINHEITEN-TREIBER
440
LESE/SCHREIB-LAUFWERK ODER SCHNITTSTELLE
432
NETZWERK-ADAPTER ODER SCHNITTSTELLE
436
434
442
444
TRAGBARE PHYSISCHE SPEICHER-EINHEIT(EN)
438

FIG. 4

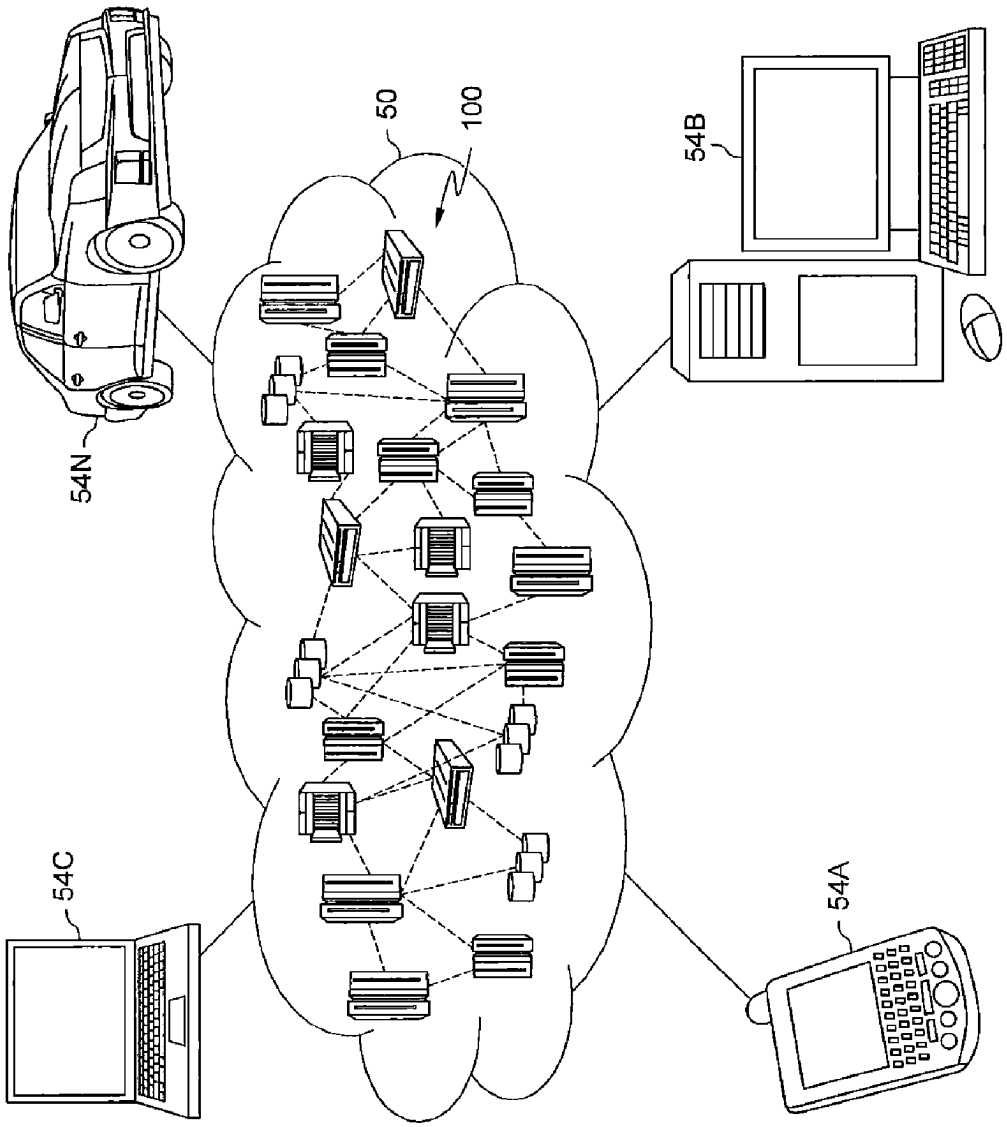


FIG. 5

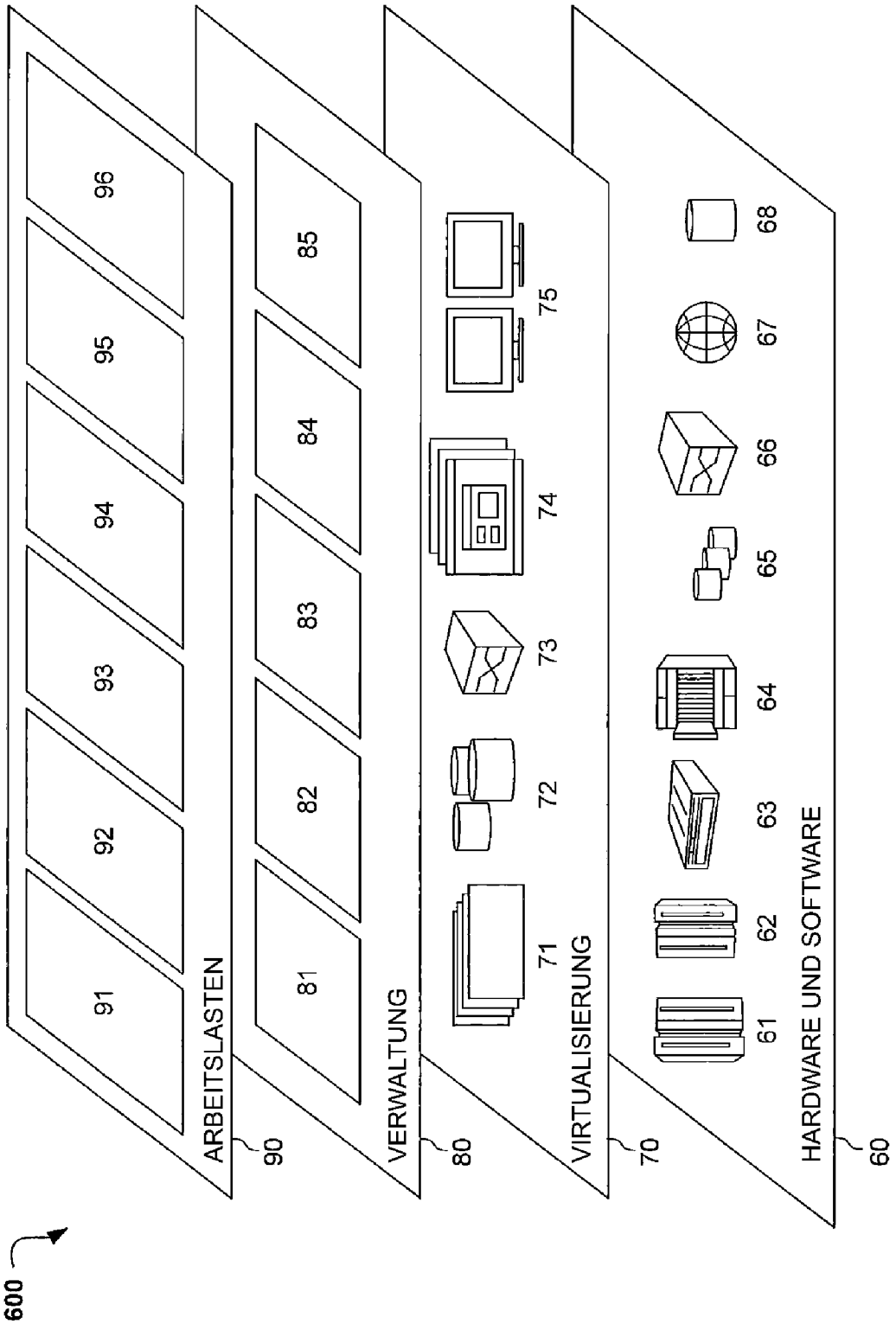

600
ARBEITSLASTEN
90
VERWALTUNG
80
VIRTUALISIERUNG
70
HARDWARE UND SOFTWARE
60
91
92
93
94
95
96
81
82
83
84
85
71
72
73
74
75
61
62
63
64
65
66
67
68

FIG. 6