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

(10) **DE 601 03 088 T2** 2004.09.09

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 117 229 B1**
(21) Deutsches Aktenzeichen: **601 03 088.5**
(96) Europäisches Aktenzeichen: **01 300 197.9**
(96) Europäischer Anmeldetag: **10.01.2001**
(97) Erstveröffentlichung durch das EPA: **18.07.2001**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **06.05.2004**
(47) Veröffentlichungstag im Patentblatt: **09.09.2004**

(51) Int Cl.[7]: **H04L 29/06**

(30) Unionspriorität:
**480788** **10.01.2000** **US**

(84) Benannte Vertragsstaaten:
**DE, FR, GB**

(73) Patentinhaber:
**Sun Microsystems, Inc., Palo Alto, Calif., US**

(74) Vertreter:
**Flaccus, R., Dipl.-Chem. Dr.rer.nat., Pat.-Anw., 50389 Wesseling**

(72) Erfinder:
**Mani, Mahalingam, Sunnyvale, US; Ramamoorthi, Sankar, San Jose, US; Mankude, Hariprasad, Fremont, US; Modi, Sohrab, Oakland, US; Fox, Kevin, San Jose, US**

(54) Bezeichnung: **Verfahren zur Herstellung von Weiterleitungslisten für Netzwerkgruppe**

## Beschreibung

### ALLGEMEINER STAND DER TECHNIK

[0001] Die vorliegende Erfindung betrifft geclusterte Computersysteme mit mehreren Netzknoten, die Dienste auf skalierbare Weise bereitstellen. Genauer gesagt betrifft die vorliegende Erfindung ein Verfahren und eine Vorrichtung, die Weiterleitungslisten bestehender Verbindungen zum Weiterleiten von Paketen an Server-Netzknoten eines Clusters verwenden.

[0002] Das jüngste explosive Wachstum des elektronischen Kommerzes hat zu einer starken Vermehrung von Websites im Internet geführt, die diverse Produkte verkaufen, wie zum Beispiel Spielzeug, Bücher und Autos, und Dienste bereitstellen, wie zum Beispiel Versicherung und Börsenhandel. Millionen Verbraucher surfen zur Zeit durch Websites, um Informationen zu sammeln, Einkäufe zu tätigen oder um unterhalten zu werden.

[0003] Der zunehmende Verkehr im Internet führt häufig zu einer gewaltigen Belastung der Server, die Hosts für Websites sind. Bestimmte beliebte Websites erhalten über eine Million „Treffer" pro Tag. Um so viel Verkehr zu verarbeiten, ohne Web-Surfer ärgerlichen Verzögerungen beim Abrufen von Websites zu unterwerfen, ist es notwendig, den Verkehr zwischen mehreren Server-Netzknoten zu verteilen, so dass mehrere Server-Netzknoten parallel arbeiten können, um den Verkehr zu verarbeiten.

[0004] Bei der Entwicklung eines solchen Systems zum verteilen von Verkehr zwischen mehreren Server-Netzknoten ist eine Anzahl von Eigenschaften wünschenswert. Es ist wünschenswert, dass ein solches System effizient ist, um so viel Verkehr wie möglich mit einer minimalen Ansprechzeit zu ermöglichen. Es ist wünschenswert, dass ein solches System „skalierbar" ist, so dass zusätzliche Server-Netzknoten hinzugefügt werden können und ein Ausgleich zwischen den Netzknoten modifizierbar ist, um einen Dienst bereitzustellen, wonn die Nachfrage nach dem Dienst zunimmt. Dabei ist es wichtig sicherzustellen, dass die Ansprechzeit nicht erhöht wird, wenn zusätzliche Server-Netzknoten hinzugefügt werden. Außerdem ist es wünschenswert, dass ein solches System konstant verfügbar ist, auch wenn einzelne Server-Netzknoten oder Kommunikationswege zwischen Server-Netzknoten ausfallen.

[0005] Ein System, das Verkehr zwischen mehreren Server-Netzknoten verteilt, führt in der Regel eine Anzahl von Tasks durch. Beim Empfang eines Pakets führt das System. ein Nachschlagen durch, um festzustellen, ob der Dienst, für den das Paket bestimmt ist, ein skalierbarer Dienst ist.

[0006] Nachdem festgestellt wurde, daß der Dienst ein skalierbarer Dienst ist, verteilt das System die Arbeitslast, die bei der Bereitstellung des Dienstes auftritt, zwischen den Server-Netzknoten, die den Dienst bereitstellen können. Es werden ein Verfahren und eine Vorrichtung zum Verteilen der Arbeitslast zwischen Server-Netzknoten mit Effizienz, Skalierbarkeit und hoher Verfügbarkeit und mit der Möglichkeit von Client-Affinität benötigt.

[0007] Nachdem ein Server-Netzknoten ausgewählt wurde, wird das Paket an den Server-Netzknoten weitergeleitet. Die herkömmliche Technik der Verwendung eines Remote Procedure Call (RPC) oder eines Aufrufs der Schnittstellendefinitionssprache (IDL) zum Weiterleiten eines Pakets umfasst in der Regel das Durchqueren eines Stapels des Internetprotokolls (IP) von einem RPC/IDL-Endpunkt zu einem Transporttreiber auf der Absenderseite und ein anschließendes Durchqueren eines weiteren IP-Stapels auf der Empfängerseite von einem Transporttreiber zu einem RPC/IDL-Endpunkt. Man beachte, dass das Durchqueren dieser beiden IP-Stapel sehr ineffizient ist. Es werden ein Verfahren und eine Vorrichtung zum Weiterleiten von Paketen zu Server-Netzknoten mit Effizienz, Skalierbarkeit und hoher Verfügbarkeit benötigt.

[0008] Es ist wünschenswert, über einen skalierbaren Dienst zu verfügen, der für eine Anwendung transparent ist. Durch diese Transparenz kann man eine Anwendung schreiben, die auf einem skalierbaren Dienst oder einem nichtskalierbaren Dienst ablaufen kann. Eine solche Anwendung ist in der Regel leichter zu schreiben, da sie die Skalierbarkeit nicht berücksichtigen muss. Ein skalierbarer Dienst, der für eine Client-Anwendung transparent ist, wäre außerdem tendenziell dazu in der Lage, bestehende Client-Anwendungen zu bereutzen. Wenn sie solche Anwendungen ausführen, können skalierbare Netzwerke die Anwendung auf einem Netzknoten des skalierbaren Dienstes ausführen. Wenn eine Reihe von Verbindungen zwischen dem Server und dem Client erforderlich ist, wäre eine Methode, dies durchzuführen, dass die Knoten in dem skalierbaren Dienst einen gemeinsam benutzten Speicher aufweisen, so dass, wenn die Client-Nachrichten zu verschiedenen Netzknoten gehen würden, jeder Netzknoten in dem System in der Lage wäre, durch Zugreifen auf den gemeinsam benutzten Speicher die Nachricht zu verarbeiten. Das gemeinsame Benutzen von Speicher verlangsamt manchmal das System und kann umständlich sein. Aus diesen Gründen wäre es wünschenswert, wenn alle Pakete von einem Client für eine Verbindung zu demselben Netzknoten in einem skalierbaren System gehen (Client-Affinität). wenn sich die Arbeitsverteilung zwischen den Netzknoten ändert, wäre es wünschenswert, wenn Pakete einer bestehenden Verbindung weiter zu demselben Netzknoten gehen, bis die Verbindung beendet wird.

[0009] Es ist wünschenswert, die Möglichkeit bereitzustellen, Pakete einer bestehenden Verbindung auch dann zu demselben Netzknoten zu senden, wenn die Arbeitslast auf einem Solaris™-Betriebssystem umver-

teilt wird, das Clustern und skalierbaren Dienst bereitstellt. Solaris™ wird von der Firma Sun Microsystems in Palo Alto, Kalifornien, hergestellt.

[0010] HUNT, G. D. H. et al.: "Network Dispatcher: a connection router for scaleable Internet Service", Computer Networks and ISDN, North Holland Publishing, Amsterdam, NL, Band 30, Nr. 1–7, 1.4.1998, Seiten 347–357, XP004121412 ISSN: 01697552, beschreibt einen TCP-Verbindungsrouter, der Lastverteilung über mehrere TCP-Server durch Überwachen der Last auf den Servern und Steuern des Verbindungszuteilungsalgorithmus in der Kernerweiterung unterstützt. Das System führt eine Verbindungstabelle und Informationen bezüglich des Status jeder Verbindung. Für jedes Paket, das ein SYN enthält, wird, wenn ein Server verfügbar ist, ein Verbindungstabelleneintrag erzeugt, der die IP-Adresse des gewählten Servers und einen Zeitstempel enthält. Ein Verbindungsstatus wird aktiv, wenn ein Paket, dessen TCP-Kopfteil SYN enthält, empfangen wird. Ein Verbindungsstatus wird beendet, wenn ein Paket, dessen TCP-Kopfteil FIN enthält, empfangen wird. Eine Verbindung wird entleert, wenn ein Paket, dessen TCP-Kopfteil RST enthält, empfangen wird. Andernfalls werden Pakete nicht verarbeitet. Client-Affinität unter FTP wird bereitgestellt, indem es einem VEC (Virtual Encapsulated Cluster) ermöglicht wird, Client-Verbindungen auf jedem Ethereal-Server-Port anzunehmen, solange eine offene Befehlsverbindung von demselben Client besteht, wie dies aus der Verbindungstabelle bestimmt wird, die angibt, dass der Client eine bestehende Verbindung mit dem Port 21 eines Servers aufweist. Client-Affinität unter SSL wird bereitgestellt, indem alte Verbindungen auf Server-Ports aufgezeichnet werden, die als „sticky" gekennzeichnet werden, und wann sie geschlossen wurden. Wenn eine neue Verbindungsanforderung für denselben Port durch denselben Client innerhalb einer Affinitätslebensspanne empfangen wird, dann wird die neue Verbindung zu demselben Server gesendet.

[0011] Die vorliegende Erfindung liefert ein System, das Weiterleitungslisten verwendet, so dass, wenn die Arbeitslast zwischen Netzknoten umverteilt wird, Pakete von einer bestehenden Verbindung weiter zu demselben Server-Netzknoten gelenkt werden, bis die Verbindung beendet wird.

[0012] Gemäß der vorliegenden Erfindung wird ein Verfahren zur Verteilung von Paketen an Server-Netzknoten in einem Netzknoten-Cluster mit den folgenden Schritten bereitgestellt: Empfangen eines Pakets an einem Schnittstellennetzknoten im Netzknoten-Cluster, wobei das Paket eine Zieladresse und eine Quelladresse einschließt, Anpassen des Pakets an ein Dienstobjekt, das einem Dienst zugeordnet ist, unter Verwendung der Zieladresse, wobei der Dienst durch die Zieladresse bestimmt wird, Abbilden der Quelladresse in einem Speicherbereich einer Mehrzahl von Speicherbereichen in einer Paketvermittlungsliste, wobei jeder Speicherbereich einen Identifikator für einen der Server-Netzknoten im Netzknoten-Cluster enthält, wobei der Speicherbereich dem an das Paket angepaßten Dienstobjekt zugeordnet ist, Feststellen, ob dem Speicherbereich eine Weiterleitungsliste zugeordnet ist, wenn festgestellt wurde, dass dem Speicherbereich eine Weiterleitungsliste zugeordnet ist, Feststellen, ob die Quelladresse mit einem Eintrag in der Weiterleitungsliste übereinstimmt, wenn es eine Übereinstimmung mit einem Eintrag in der Weiterleitungsliste gibt, Senden des Pakets an den Server-Netzknoten, auf den durch die Übereinstimmung mit dem Eintrag in der Weiterleitungsliste verwiesen wird, wodurch sichergestellt wird, dass eine bereits bestehende Verbindung nicht unterbrochen wird, und wenn es keine Übereinstimmung mit einem Eintrag in der Weiterleitungsliste gibt, Senden des Pakets an den Server-Netzknoten, auf den durch den Speicherbereich verwiesen wird, in dem die Quelladresse des Pakets abgebildet ist.

[0013] Gemäß weiteren Aspekten der Erfindung wird ein computerlesbares Medium bereitgestellt, das Computeranweisungen wie in Anspruch 1 speichert, die, wenn sie ausgeführt werden, das erfindungsgemäße Verfahren durchführen, sowie eine Vorrichtung nach Anspruch 1 zum Ausführen des erfindungsgemäßen Verfahrens.

[0014] Diese und andere Merkmale der vorliegenden Erfindung werden im folgenden in der ausführlichen Beschreibung der Erfindung und in Verbindung mit den folgenden Figuren ausführlicher beschrieben.

[0015] Die vorliegende Erfindung wird in den Figuren der beigefügten Zeichnungen, in denen gleiche Bezugszahlen ähnliche Elemente bedeuten, beispielhaft und nicht einschränkend dargestellt. Es zeigen:

[0016] **Fig.** 1 ein geclustertes Datenverarbeitungssystem, das durch ein Netzwerk mit Client-Datenverarbeitungssystemen gekoppelt ist, gemäß einer Ausführungsform der vorliegenden Erfindung;

[0017] **Fig.** 2 die interne Struktur eines Schnittstellennetzknotens und zweier Server-Netzknoten in einem geclusterten Datenverarbeitungssystem gemäß einer Ausführungsform der vorliegenden Erfindung;

[0018] **Fig.** 3 einem skalierbaren Dienst zugeordnete Datenstrukturen gemäß einer Ausführungsform der vorliegenden Erfindung;

[0019] **Fig.** 4, wie ein IP-Paket gemäß einer Ausführungsform der vorliegenden Erfindung mit einem DLPI-Kopfteil verkapselt wird;

[0020] **Fig.** 5A ein Flussdiagramm des Prozesses der Dienstregistration gemäß einer Ausführungsform der vorliegenden Erfindung;

[0021] **Fig.** 5B ein Flussdiagramm des Prozesses der Dienstaktivierung/-Reaktivierung gemäß einer Ausführungsform der vorliegenden Erfindung;

[0022] **Fig.** 6 ein Flussdiagramm, das zeigt, wie ein Paket gemäß einer Ausführungsform der vorliegenden

Erfindung in einem Schnittstellennetzknoten verarbeitet wird;
[0023] **Fig.** 7 ein Flussdiagramm des Prozesses des Nachschlagens eines Dienstes für ein Paket gemäß einer Ausführungsform der vorliegenden Erfindung;
[0024] **Fig.** 8 ein Flussdiagramm des Prozesses des Weiterleitens eines Pakets an einen Server gemäß einer Ausführungsform der vorliegenden Erfindung;
[0025] **Fig.** 9, wie ein PDT-Server gemäß einer Ausführungsform der vorliegenden Erfindung im Checkpoint-Verfahren mit einem Slave-PDT-Server und einem sekundären PDT-Server verknüpft wird;
[0026] **Fig.** 10 ein Flussdiagramm des Aktualisierens einer Weiterleitungsliste gemäß einer Ausführungsform der vorliegenden Erfindung.

## AUSFÜHRLICHE BESCHREIBUNG DER BEVORZUGTEN AUSFÜHRUNGSFORMEN

[0027] Die vorliegende Erfindung wird nun im einzelnen mit Bezug auf einige wenige bevorzugte Ausführungsformen und wie in den beigefügten Zeichnungen dargestellt beschrieben. In der folgenden Beschreibung werden zahlreiche spezifische Einzelheiten dargelegt, um ein eingehendes Verständnis der vorliegenden Erfindung zu ermöglichen. Für Fachleute ist jedoch erkennbar, dass die vorliegende Erfindung ohne einen Teil dieser spezifischen Einzelheiten oder alle diese ausgeübt werden kann. In anderen Fällen wurden wohlbekannte Verfahrensschritte und/oder Strukturen nicht im Einzelnen beschrieben, um die vorliegende Erfindung nicht unnötig zu verkomplizieren.
[0028] Die in der vorliegenden ausführlichen Beschreibung beschriebenen Datenstrukturen und der in der vorliegenden ausführlichen Beschreibung beschriebene Code werden in der Regel auf einem computerlesbaren Speichermedium gespeichert, bei dem es sich um eine beliebige Einrichtung oder ein beliebiges Medium handeln kann, die bzw. das Code und/oder Daten zur Verwendung durch ein Computersystem speichern kann. Dazu gehören ohne Einschränkung magnetische und optische Speichereinrichtungen, wie zum Beispiel Plattenlaufwerke, Magnetband, CDs (Compact Discs) und DVDs (Digital Video Discs), RAM (Direktzugriffsspeicher) und Computeranweisungssignale, die in einem Übertragungsmedium (mit oder ohne eine Trägerwelle, auf die die Signale aufmoduliert werden) realisiert werden. Zum Beispiel kann das Übertragungsmedium ein Kommunikationsnetzwerk, wie zum Beispiel das Internet, enthalten.

### Geclustertes Datenverarbeitungssystem

[0029] **Fig.** 1 zeigt ein geclustertes Datenverarbeitungssystem **100**, das gemäß einer Ausführungsform der vorliegenden Erfindung durch Netzwerke **120** an Clients **121**–**123** angekoppelt ist. Die Clients **121**–**123** können einen beliebigen Netzknoten in den Netzwerken **120** enthalten, einschließlich Rechenfähigkeit und einschließlich eines Mechanismus zur Kommunikation über die Netzwerke **120**. Die Clients **121**–**123** kommunizieren mit dem geclusterten Datenverarbeitungssystem **100** durch Senden von Paketen zu dem geclusterten Datenverarbeitungssystem **100**, um Dienste von dem geclusterten Datenverarbeitungssystem **100** anzufordern.
[0030] Die Netzwerke **120** können eine beliebige Art von drahtgebundenem oder drahtlosem Kommunikationskanal einschließen, der Datenverarbeitungs-Netzknoten miteinander koppeln kann. Dazu gehören ohne Einschränkung ein lokales Netzwerk, ein großflächiges Netzwerk oder eine Kombination von Netzwerken. Bei einer Ausführungsform der vorliegenden Erfindung enthalten die Netzwerke **120** das Internet.
[0031] Das geclusterte Datenverarbeitungssystem **100** enthält eine Menge von Netzknoten, die durch einen (nicht gezeigten) Kommunikationskanal miteinander gekoppelt sind. Diese Knoten schließen Server-Netzknoten **102** und **104** sowie den Schnittstellennetzknoten/Servernetzknoten **103** ein. Die Knoten **102**–**104** sind an das Speichersystem **110** angekoppelt. Das Speichersystem **110** liefert Archivspeicherung für Code und/oder Daten, der bzw. die durch die Netzknoten **102**–**104** manipuliert werden, bereit. Zu dieser Archivspeicherung kann ohne Einschränkung magnetische Speicherung, optische Speicherung, Flash-Speicher, ROM, EPROM, EEPROM und RAM mit Batteriesicherung gehören.
[0032] Die Netzknoten **102**–**104** sind durch eine private Verbindung mit (nicht gezeigten) redundanten Pfaden miteinander gekoppelt. Zum Beispiel können die Netzknoten durch einen Kommunikationsmechanismus miteinander verbunden werden, der den Normen des Ethernets oder SCI (Scalable Coherent Interconnect) genügen. Ein Wegmanager operiert an allen Netzknoten in dem geclusterten Datenverarbeitungssystem **100**. Dieser Wegmanager kennt die Verbindungstopologie und überwacht den Status von Pfaden. Außerdem stellt der Wegmanager eine Schnittstellenregistrierdatenbank bereit, bei der sich andere an dem Status der Verbindung interessierte Komponenten registrieren lassen können. Dadurch wird ein Mechanismus für den Wegmanager bereitgestellt, um Callbacks an die interessierten Komponenten durchzuführen, wenn sich der Status eines Weges ändert, wenn sich ein neuer Weg anbietet, oder wenn ein Weg entfernt wird.
[0033] Die Knoten **102**–**104** sind durch ein hochverfügbares Adressierungssystem **108** an die Netzwerke **120** angekoppelt. Das hochverfügbare Adressierungssystem **108** ermöglicht eine Adressierung von Schnittstellennetzknoten in dem geclusterten Datenverarbeitungssystem **100** von dem Netzwerk **120** auf auf „hochverfüg-

bare Weise", so dass, wenn ein Schnittstellennetzknoten ausfällt, ein sekundärer Reserveschnittstellennetzknoten eine Stelle einnehmen kann, ohne dass der Ausfall für die Clients **121**–**123** sichtbar ist. Man beachte, dass der Schnittstellennetzknoten **103** Host für eine oder mehrere gemeinsam benutzte IP-Adressen für das geclusterte Datenverarbeitungssystem **100** sein kann. Man beachte außerdem, dass mehr als ein Netzknoten in dem geclusterten Datenverarbeitungssystem **100** als ein Reserveschnittstellennetzknoten für einen gegebenen Dienst wirken kann. Dadurch kann ein Reserveschnittstellennetzknoten von einem ausfallenden Schnittstellennetzknoten übernehmen.

[0034] Man beachte, dass die Netzknoten **102**–**104** in dem geclusterten Datenverarbeitungssystem **100** skalierbare Dienste bereitstellen können. Jeder skalierbare Dienst verhält sich vom Standpunkt der Clients **121**–**123** als eine einzige logische Entität. Außerdem beachte man, dass die Clients **121**–**123** durch eine TCP-Verbindung (Transmission Control Protocol) oder eine UDP-Session (User Datagram Protocol) mit dem geclusterten Datenverarbeitungssystem **100** kommunizieren können.

[0035] Wenn die Last auf einem Dienst zunimmt, versucht der Dienst, dieselbe Ansprechzeit pro Client aufrechtzuerhalten. Ein Dienst wird als „skalierbar" bezeichnet, wenn eine erhöhte Last auf dem Dienst mit einer Zunahme von Hardware- und Server-Instanzen, die den Dienst durchführen, einhergeht oder mit einer Veränderung der Ausgleichsverteilung zwischen Netzknoten. Zum Beispiel ist ein Web-Server skalierbar, wenn eine zusätzliche Last auf dem Web-Server mit einer entsprechenden Zunahme der Server-Netzknoten zur Verarbeitung der zusätzlichen Last einhergeht oder mit einer Änderung der Verteilung der Last über die Hardware- und Server-Instanzen, die den Dienst durchführen.

[0036] Das geclusterte Datenverarbeitungssystem **100** arbeitet im allgemeinen folgendermaßen. Während Pakete an dem Schnittstellennetzknoten **103** von den Clients **121**–**123** ankommen, wird auf der Basis der Zieladresse in dem Paket ein Dienst für das Paket ausgewählt. Als nächstes wird für das Paket auf der Basis der Quellenadresse des Pakets sowie der Zieladresse des Pakets eine Server-Instanz ausgewählt. Man beachte, dass das System sicherstellt, dass Pakete, die zu derselben TCP-Verbindung gehören, zu derselben Server-Instanz gesendet werden. Als letztes wird das Paket zu der gewählten Server-Instanz gesendet.

## Interne Struktur von Schnittstellennetzknoten und Server-Netzknoten

[0037] **Fig.** 2 zeigt die interne Struktur des Schnittstellennetzknotens und der Server-Netzknoten **102** und **104** in dem geclusterten Datenverarbeitungssystem **100** (**Fig.** 1) gemäß einer Ausführungsform der vorliegenden Erfindung. Der Client **121** sendet Pakete zu dem geclusterten Datenverarbeitungssystem **100**, um einen Dienst von dem geclusterten Datenverarbeitungssystem **100** zu erhalten. Diese Pakete treten in die öffentliche Schnittstelle **221** in dem Schnittstellennetzknoten in dem geclusterten Datenverarbeitungssystem **100** ein. Die öffentliche Schnittstelle **221** kann eine beliebige Art von Schnittstelle enthalten, die Pakete von den Netzwerken **120** empfangen kann.

[0038] Während Pakete über die öffentliche Schnittstelle **221** an dem Schnittstellennetzknoten ankommen, werden sie durch den Cluster-Vernetzungsmultiplexer **218** geleitet. Der Cluster-Vernetzungsmultiplexer **218** leitet die Pakete auf der Basis von Lastausgleichsrichtlinien und anderen Gesichtspunkten zu verschiedenen Netzknoten in dem geclusterten Datenverarbeitungssystem **100** weiter. Beim Treffen von Weiterleitungsentscheidungen ruft der Cluster-vernetzungsmultiplexer **218** Daten von dem hochverfügbaren PDT-Server **230** ab. Die Struktur dieser Daten wird nachfolgend ausführlicher in Bezug auf **Fig.** 3 beschrieben. Man beachte, dass der HA-PDT-Server **230** über mehrere Netzknoten des geclusterten Datenverarbeitungssystems **100** hinweg dupliziert werden kann, so dass im Fall des Ausfalls eines Knotens ein Reservenetzknoten für ihn übernehmen kann, um die Verfügbarkeit für den HA-PDT-Server **230** aufrechtzuerhalten.

[0039] Pakete werden von dem Schnittstellennetzknoten zu anderen Netzknoten in dem geclusterten Datenverarbeitungssystem **100**, einschließlich der Server-Netzknoten **102** und **104**, durch die privaten Schnittstellen **224** und **225** weitergeleitet. Die privaten Schnittstellen **224** und **225** können jede beliebige Schnittstelle einschließen, die die Kommunikation zwischen Netzknoten in dem geclusterten Datenverarbeitungssystem **100** abwickeln kann. Zum Beispiel können Pakete aus der privaten Schnittstelle **224** zu der privaten Schnittstelle **226** auf dem Server-Netzknoten **104** oder von der privaten Schnittstelle **225** zu der privaten Schnittstelle **228** auf dem Server-Netzknoten **102** weitergeleitet werden. Man beachte, dass die privaten Schnittstellen **224** und **225** keine Kommunikation mit Entitäten außerhalb des geclusterten Datenverarbeitungssystems **100** abwickeln.

[0040] Bei bestimmten Ausführungsformen der vorliegenden Erfindung benutzen die private Schnittstelle **224** (und **225**) und die öffentliche Schnittstelle **221** einen Teil derselben Kommunikationshardware gemeinsam und senden Nachrichten entlang eines Teils derselben physischen Datenwege. Bei einigen dieser Ausführungsformen können die private Schnittstelle **224** und die öffentliche Schnittstelle **221** außerdem einen Teil derselben Schnittstellensoftware gemeinsam benutzen. Daher müssen die private Schnittstelle **224** und die öffentliche Schnittstelle **221** nicht unbedingt verschiedene Kommunikationsmechanismen darstellen. Deshalb kann die Unterscheidung zwischen der privaten Schnittstelle **224** und der öffentlichen Schnittstelle **221** lediglich dazwi-

schen unterscheiden, ob die Kommunikation mit einer Entität außerhalb des geclusterten Datenverarbeitungssystems **100** oder mit einer Entität in dem geclusterten Datenverarbeitungssystem **100** stattfindet.

[0041] Pakete, die in die Server-Netzknoten **102** und **104** eintreten, werden durch IP-Stapel **214** bzw. **216** geleitet. Der Cluster-Vernetzungsmultiplexer **218** kann außerdem Pakete zu dem IP-Stapel **215** in dem Schnittstellennetzknoten/Server-Netzknoten senden, weil der Netzknoten auch als ein Server wirken kann. Auf dem Server-Netzknoten **102** werden Pakete durch den IP-Stapel **214** in das TCP-Modul **206** geleitet, das TCP-Verbindungen unterstützt, oder in das UDP-Modul **210**, das UDP-Sessions unterstützt. Ähnlich werden auf. dem Schnittstellennetzknoten/Server-Netzknoten Pakete durch den IP-Stapel **215** in das TCP-Modul **207** oder in das UDP-Modul **211** geleitet. Auf dem Server-Netzknoten **104** werden Pakete durch den IP-Stapel **216** in das TCP-Modul **208** oder in das UDP-Modul **212** geleitet. Als nächstes werden die Pakete durch Server-Instanzen **201**–**203** jeweils auf den Netzknoten **102**–**104** verarbeitet.

[0042] Man beachte, dass die Rückkehrkommunikation für die Server-Netzknoten **102** und **104** nicht demselben Weg folgt. Die Rückkehrkommunikation von dem Server-Netzknoten **102** wird durch den IP-Stapel **214**, durch die öffentliche Schnittstelle **220** und dann zu dem Client **121** geleitet. Ähnlich wird die Rückkehrkommunikation von dem Server-Netzknoten **104** durch den IP-Stapel **216**, durch die öffentliche Schnittstelle **222** und dann zu dem Client **121** geleitet. Dadurch wird der Schnittstellennetzknoten davon befreit, Rückkehrkommunikationsverkehr abwickeln zu müssen.

[0043] Bei Web-Server- oder anderen Anwendungen kann dieser Rückkehrkommunikationsmechanismus Lastausgleich für Rückkehrverkehr bereitstellen. Man beachte, dass Web-Server in der Regel Navigationsbefehle von einem Client empfangen und als Reaktion große Volumen an Webseiteninhalt (wie zum Beispiel grafische Bilder) zu dem Client zurücksenden. Für diese Anwendungen ist es vorteilhaft, den Rückkehrverkehr über mehrere Rückkehrpfade zu verteilen, um das große Volumen an Rückkehrverkehr abzuwickeln.

[0044] Man beachte, dass in einem Server-Netzknoten, wie zum Beispiel einem Proxy-Netzknoten (ein nichtglobaler Schnittstellennetzknoten für diese IP-Adresse), wie zum Beispiel dem Server-Netzknoten **104**, gemeinsam benutzte IP-Adressen auf der „Loopback-Schnittstelle" des Server-Netzknotens **104** gehostet werden. (Die Loopback-Schnittstelle wird in den Betriebssystemstandards für UNIX und SOLARIS™ definiert. Solaris™ ist ein Warenzeichen der Firma Sun Microsystems in Palo Alto, Kalifornien).

## Datenstrukturen zur Unterstützung skalierbarer Dienste

[0045] **Fig.** 3 zeigt einem skalierbaren Dienst zugeordnete Datenstrukturen gemäß einer Ausführungsform der vorliegenden Erfindung. Der HA-PDT-Server **230** enthält mindestens eine Dienstgruppe **302**. Die Dienstgruppe **302** kann einer Gruppe von Dienstobjekten (identifiziert durch eine eindeutige Kombination von Protokoll, Dienst-IP-Adresse und Dienstportnummer), die eine gemeinsame Lastausgleichrichtlinie benutzen, zugeordnet sein.

[0046] Man beachte außerdem, dass die Dienstgruppe **302** mindestens eine zugeordnete sekundäre Version auf einem anderen Netzknoten und mögliche Ersatzteile für Fehlertoleranzzwecke aufweisen kann. Jegliche Änderungen an der Dienstgruppe **302** können im Checkpoint-Verfahren mit dieser sekundären Version verknüpft werden, so dass, wenn der Netzknoten, der die primäre Version der Dienstgruppe **302** enthält, ausfällt, der Netzknoten, der die sekundäre Version enthält, übernehmen kann.

[0047] Die Dienstgruppe **302** kann außerdem einer Anzahl von „Slave"-Versionen der auf anderen Netzknoten in dem geclusterten Datenverarbeitungssystem **100** angeordneten Dienstgruppe zugeordnet sein. Dadurch können die anderen Netzknoten schneller auf die Daten in der Dienstgruppe **302** zugreifen. Jegliche Änderung an der Dienstgruppe **302** kann zu den entsprechenden Slave-Versionen propagiert werden. SlavVersionen können Ableitungen der Master-Dienstgruppe sein, die nicht alle Informationen in der Master-Dienstgruppe besitzen.

[0048] Die Dienstgruppe **302** enthält eine Anzahl von Datenstrukturen, einschließlich der Verteilungstabelle (PDT) **304**, der Lastausgleichsrichtlinie **306**, eines Dienstobjekts **308**, der Konfigurationsnetzknotenliste **310** und der Instanznetzknotenliste **312**.

[0049] Die Konfigurationsnetzknotenliste **310** enthält eine Liste von Server-Netzknoten in dem geclusterten Datenverarbeitungssystem **100**, die die der Dienstgruppe **302** zugeordneten Dienste bereitstellen können. Das Dienstobjekt **308** erhält Konfigurationsdaten von der Konfigurationsnetzknotenliste **310**. Die Instanznetzknotenliste **312** enthält eine Liste der Netzknoten, die tatsächlich zur Bereitstellung dieser Dienste benutzt werden. Das Dienstobjekt **308** enthält Informationen bezüglich eines Dienstes (identifiziert durch eine eindeutige Kombination von Protokoll, Dienst-IP-Adresse und Dienstportnummer), der der Dienstgruppe **302** zugeordnet ist. Es kann mehr als ein Dienstobjekt in jeder Dienstgruppe geben. Jeder Dienst (identifiziert durch eine eindeutige Kombination von Protokoll, Dienst-IP-Adresse und Dienstportnummer), der der Dienstgruppe zugeordnet ist, ist einem Dienstobjekt (identifiziert durch dieselbe eindeutige Kombination von Protokoll, Dienst-IP-Adresse und Dienstportnummer wie der zugeordnete Dienst) in der Dienstgruppe zugeordnet.

[0050] Die Lastausgleichsrichtlinie **306** enthält eine Beschreibung einer Lastausgleichsrichtlinie, die zum Ver-

teilen von Paketen zwischen an der Bereitstellung von der Dienstgruppe **302** zugeordneten Diensten beteiligten Netzknoten verwendet wird. Die Lastausgleichsrichtlinie **306** kann den Lastausgleichsrichtlinientyp und Lastausgleichsrichtliniengewichte spezifizieren. Die beschriebene Ausführungsform der Erfindung verwendet mindestens drei Arten von Lastausgleichsrichtlinien, einschließlich 1) einer Nicht-Affinitäts-Richtlinie, 2) einer Client-Affinität-Richtlinie und 3) einer Wildcard-Client-Affinität-Richtlinie. Es kann verschiedene Arten von Nicht-Affinität-Richtlinien geben, wie zum Beispiel eine gewichtete oder Reigen-Lastausgleichs-Richtlinie. Die Gewichte können spezifizieren, dass ein bestimmter Prozentsatz an Verkehr zu einem bestimmten Netzknoten gesendet wird.

[0051] Die PDT **304** dient zum Implementieren der Lastausgleichsrichtlinie. Die PDT **304** enthält Einträge, die mit Identifikatoren für Netzknoten angefüllt sind, die zur Zeit in der Lage sind, Pakete für die der Dienstgruppe **302** zugeordneten Dienste zu empfangen. Um einen Server-Netzknoten auszuwählen, zu dem ein Paket weitergeleitet werden soll, verarbeitet das System die Client-Adresse und potenziell den Client-Port gemäß der Lastausgleichsrichtlinie der PDT **304** im Hash-Verfahren, wie später ausführlicher beschrieben wird. Diese Hash-Verarbeitung wählt einen bestimmten Eintrag in der PDT **304** und dieser Eintrag verweist auf einen Server-Netzknoten in dem geclusterten Datenverarbeitungssystem **100**.

[0052] Man beachte, dass jede beliebige Zufalls- oder Pseudozufallsfunktion zur Hash-Verarbeitung der Quellenadresse verwendet werden kann.

[0053] Man beachte außerdem, dass die Häufigkeit von Einträgen variiert werden kann, um verschiedene Verteilungen von Verkehr zwischen verschiedenen Server-Netzknoten zu erreichen. Zum Beispiel können einem Hochleistungs-Server-Netzknoten, der eine große Menge an Verkehr verarbeiten kann, mehr Einträge in der PDT **304** gegeben werden als einem langsameren Server-Netzknoten, der weniger Verkehr verarbeiten kann. Auf diese Weise erhält der Hochleistungs-Server-Netzknoten im Mittel mehr Verkehr als der langsamere Server-Netzknoten.

[0054] Man beachte außerdem, dass, wenn ein PDT-Server mit Konfigurationsdaten in seinem lokalen Speicher ausfällt, ein sekundärer PDT-Server übernimmt. Der Checkpoint-Prozess stellt sicher, dass die Konfigurationsdaten ebenfalls in dem lokalen Speicher für den sekundären PDT-Server vorhanden sein werden. Genauer gesagt zeigt **Fig.** 9, wie ein PDT-Server im Checkpoint-Verfahren mit einem Slave-PDT-Server und einem sekundären PDT-Server gemäß einer Ausführungsform der vorliegenden Erfindung verknüpft wird. wie in **Fig.** 9 gezeigt, führt das System einen primären bzw. Master-PDT-Server **912** auf dem Netzknoten **910**. Für Hochverfügbarkeitszwecke wird der Zustand des primären bzw. Master-PDT-Servers **912** im Checkpoint-Verfahren mit dem sekundären PDT-Server **904** auf dem Netzknoten **902** verknüpft, so dass der zweite PDT-Server **904** konsistent mit dem primären bzw. Master-PDT-Server **912** gehalten wird. Auf diese Weise kann, wenn der primäre bzw. Master-PDT-Server **912** ausfällt, der sekundäre PDT-Server **904** seinen Platz einnehmen.

[0055] Wenn sich der primäre bzw. Master-PDT-Server **912** nicht auf einem Schnittstellennetzknoten **906** befindet, wird auf dem Schnittstellennetzknoten **906** aus Leistungsgründen (nicht aus Hochverfügbarkeitsgründen) ein Slave-PDT-Server **908** geführt. In diesem Fall wird der größte Teil des Zustands des primären bzw. Master-PDT-Servers **912** im Checkpoint-Verfahren mit dem Slave-PDT-Server **908** in dem Schnittstellennetzknoten **906** verknüpft. Dadurch kann der Schnittstellennetzknoten **906** auf die mit der Paketweiterleitung zusammenhängenden Informationen lokal in dem Slave-PDT-Server **908** zugreifen, ohne mit dem Netzknoten-Primär- bzw. Master-PDT-Server **912** auf dem Netzknoten **910** kommunizieren zu müssen.

[0056] Master-Dienstgruppen werden in dem primären bzw. Master-PDT-Server **912** geführt, wobei die Master-Dienstgruppen Master-Dienstobjekte aufweisen. Sekundäre Kopien der Master-Dienstgruppen und Master-Dienstobjekte werden in dem sekundären PDT-Server **904** geführt. Slave-Kopien der Master-Dienstgruppe, die Teilmengen oder Ableitungen der Master-Dienstgruppe sind, und Slave-Kopien der Master-Dienstobjekte, die Teilmengen oder Ableitungen der Master-Dienstobjekte sind, werden in dem Slave-PDT-Server **908** geführt. In der Beschreibung, in der offengelegt wird, dass Daten aus einer Dienstgruppe gelesen oder in eine Dienstgruppe geschrieben werden, können sie tatsächlich in einen Master, eine sekundäre Kopie oder in eine Slave-Kopie der Master-Dienstgruppe geschrieben werden. Wenn offengelegt wird, dass Daten aus einem Dienstobjekt gelesen oder in ein Dienstobjekt geschrieben werden, kann es außerdem sein, dass sie tatsächlich in einen Master, in eine sekundäre Kopie oder in eine Slave-Kopie des Master-Dienstobjekt geschrieben werden.

## Paketweiterleitung

[0057] **Fig.** 4 zeigt, wie ein IP-Paket **400** gemäß einer Ausführungsform der vorliegenden Erfindung mit einem DLPI-Kopfteil **402** verkapselt wird. Damit ein IP-Paket **400** zwischen dem Schnittstellennetzknoten und dem Server-Netzknoten **104** (siehe **Fig.** 2) weitergeleitet werden kann, wird der DLPI-Kopfteil **402** an den Kopf des IP-Pakets **400** angefügt. Man beachte, dass der DLPI-Kopfteil **402** die MAC-Adresse (Medium Access Control) des Ziel-Server-Netzknotens **104** enthält. Man beachte außerdem, dass das IP-Paket **400** eine Zieladresse **404** enthält, die bei der bevorzugten Ausführungsform ein Protokoll, eine Dienst-IP-Adresse und eine

Dienst-Portnummer eines Dienstes, für den der Schnittstellennetzknoten der Host ist, sowie die Quellenadresse **406** spezifiziert, die bei der bevorzugten Ausführungsform eine Client-IP-Adresse und Client-Portnummer für einen Client, der das Paket gesendet hat, spezifiziert.

Konfigurationsprozess

[0058] **Fig.** 5A ist ein Flussdiagramm des Prozesses zum Konfigurieren eines Lastausgleichsystems gemäß einer Ausführungsform der vorliegenden Erfindung. Das System startet durch Konfigurieren einer Skalierbarer-Dienst-Gruppe (Schritt **501**). Dabei wird eine Dienstgruppe für die Skalierbarer-Dienst-Gruppe erzeugt (Schritt **502**) und ein Lastausgleichsrichtlinientyp für die Dienstgruppe spezifiziert (Schritt **503**). Bei der bevorzugten Ausführungsform der Erfindung gibt es mindestens drei Arten von Lastausgleichsrichtlinien, darunter 1) eine Nicht-Affinität-Richtlinie, 2) eine Client-Affinität-Richtlinie und 3) eine Wildcard-Client-Affinität-Richtlinie. Es gibt verschiedene Arten von Affinität-Richtlinien, wie zum Beispiel eine gewichtete oder Reigen-Lastausgleichsrichtlinie. Bei der bevorzugten Ausführungsform der Erfindung werden Lastausgleichsrichtliniengewichte anfangs auf einen Vorgabewert gleicher Gewichtung für jeden Netzknoten gesetzt. Die Gewichte können später im Schritt **508** verschieden gesetzt werden. Für einen bestimmten Dienst, der durch ein eindeutiges Protokoll, Dienst-IP-Adresse und Portnummer spezifiziert wird, wird ein Dienstobjekt erzeugt (Schritt **504**) und einer Dienstgruppe zugewiesen. Der durch das Dienstobjekt identifizierte Dienst wird entweder in einer ersten Dienstlisten-Hash-Tabelle oder in einer zweiten Dienstlisten-Hash-Tabelle, die alle durch alle Dienstobjekte in allen Dienstgruppen identifizierten Dienste auflistet, abgelegt. Dienste, die durch Dienstobjekte in Dienstgruppen identifiziert werden und keine Wildcard-Client-Affinität-Lastausgleichsrichtlinie aufweisen, werden in der ersten Dienstlisten-Hash-Tabelle abgelegt. Durch Dienstobjekte in Dienstgruppen mit einem Wildcard-Client-Affinität-Lastausgleichsrichtlinientyp identifizierte Dienste werden in der zweiten Dienstlisten-Hash-Tabelle abgelegt. Zusätzlich sollten Dienste mit derselben IP-Dienstadresse wie ein nichtskalierbarer Dienst nicht in einer Dienstgruppe mit Wildcard-Client-Affinität abgelegt werden. Das Konfigurieren einer Skalierbarer-Dienst-Gruppe umfasst außerdem das Initialisieren einer Konfigurationsnetzknotenliste **310** (Schritt **506**), um anzugeben, welche Server-Netzknoten in dem geclusterten Datenverarbeitungssystem **100** die Dienstgruppe bereitstellen können, und das Setzen von Lastausgleichsrichtliniengewichten **306** für die Dienstgruppe, um den Ausgleich zwischen den Netzknoten von der Vorgabeeinstellung abzuändern (Schritt **508**). Man beachte, dass eine bestimmte Lastausgleichsrichtlinie Gewichte für die bestimmten Server-Netzknoten spezifizieren kann.

[0059] **Fig.** 5B ist ein Flussdiagramm des Prozesses der Dienstaktivierung/-deaktivierung gemäß einer Ausführungsform der vorliegenden Erfindung. Dieser Prozess findet immer dann statt, wenn eine Instanz gestartet oder gestoppt wird, oder immer dann, wenn ein Netzknoten ausfällt. Für jede Skalierbarer-Dienst-Gruppe untersucht das System jeden Netzknoten auf der Konfigurationsknotenliste **310**. Wenn der Netzknoten mit der laufenden Version der Skalierbarer-Dienst-Gruppe übereinstimmt, dann wird der Netzknoten zu der PDT **304** und zu der Instanznetzknotenliste **312** hinzugefügt (Schritt **510**).

[0060] Wenn zu einem bestimmten Zeitpunkt in der Zukünft ein Netzknoten abstürzt oder der Dienst abstürzt, wird ein entsprechender Eintrag aus der PDT **304** und der Instanznetzknotenliste **312** entfernt (Schritt **512**).

Paketverarbeitung

[0061] **Fig.** 6 ist ein Flussdiagramm, das zeigt, wie ein Paket gemäß einer Ausführungsform der vorliegenden Erfindung in einem Schnittstellennetzknoten verarbeitet wird. Das System startet, indem es das IP-Paket **400** von dem Client **121** in dem Cluster-Vernetzungsmultiplexer **218** in dem Schnittstellennetzknoten empfängt (Schritt **601**). Das IP-Paket **400** enthält eine Zieladresse **404**, die einen Dienst spezifiziert, und eine Client-Adresse **406** des Client, der das Paket gesendet hat.

[0062] Das System schlägt zunächst einen Dienst für das Paket auf der Basis der Zieladresse nach, bei der es sich bei der bevorzugten Ausführungsform um die Protokoll-, Dienst-IP-Adresse und die Dienstportnummer **404** handelt (Schritt **602**). Dieser Nachschlageprozess wird nachfolgend mit Bezug auf **Fig.** 7 ausführlicher beschrieben.

[0063] Als nächstes stellt das System fest, ob der Server ein skalierbarer Dienst ist (Schritt **603**), was im Schritt **710** von **Fig.** 7, der später ausführlicher beschrieben wird, geflaggt wird. Wenn der Dienst nichtskalierbar ist, sendet das System das Paket zu dem IP-Stapel **215** in dem Schnittstellennetzknoten/Server-Netzknoten , so dass die Server-Instanz **202** den nichtskalierbaren Dienst bereitstellen kann (Schritt **604**). Als Alternative kann der Schnittstellennetzknoten das Paket zu einem Vorgabe-Server-Netzknoten außerhalb des Schnittstellennetzknotens/Server-Netzknotens senden, um den nichtskalierbaren Dienst bereitzustellen. Zum Beispiel kann der Server-Netzknoten **104** als ein Vorgabe-Netzknoten für nichtskalierbare Dienste vermerkt werden. Ein skalierbarer Dienst ist ein Dienst, der von einem oder mehreren Netzknoten in einem Cluster und so, wie es eine Last auf dem Cluster verlangt, bereitgestellt werden kann. Ein nichtskalierbarer Dienst ist ein

Dienst, der nur auf einem Netzknoten bereitgestellt werden kann.

[0064] Wenn der Dienst ein skalierbarer Dienst ist, stellt das System fest, zu welchem Server-Netzknoten das Paket gesendet werden soll. Dabei stellt das System zunächst fest, ob die dem Dienst des Pakets zugeordnete Dienstgruppe einen Lastausgleichsrichtlinientyp mit Client-Affinität aufweist (Schritt **605**), d. h. ob der Lastausgleichsrichtlinientyp Client-Affinität oder Wildcard-Client-Affinität ist. Wenn dies der Fall ist, verarbeitet das System die Client-IP-Adresse über die PDT **304** im Hash-Verfahren, um einen Speicherbereich aus der PDT **304** zu wählen (Schritt **606**). Wenn nicht, verarbeitet das System die Client-IP-Adresse und die Port-Nummer im Hash-Verfahren, um einen Speicherbereich aus der PDT **304** auszuwählen (Schritt **607**). Es ist zu beachten, dass, wenn der Richtlinientyp eine Client-Affinität-Richtlinie ist, nur die Client-Adresse im Hash-Verfahren verarbeitet wird (statt sowohl die Client-Adresse als auch die Portnummer). Dies ist in vielen Systemen wichtig, in denen eine einzige Quelle mehrere parallele Verbindungen mit einem Server aufweisen kann, der die Informationen aus den parallelen Verbindungen kombinieren muss (zum Beispiel kann beim Anhören einer Internet-Rundsendung eine Verbindung zum Empfangen der Rundsendung und eine weitere Verbindung zur Steuerung der Rundsendung verwendet werden). Wenn Client-Affinität nicht benötigt wird, ist das Hash-Verarbeiten sowohl der Client-Adresse als auch des Client-Ports statistisch tendenziell besser für den Lastausgleich.

[0065] Als nächstes stellt das System fest, ob das Protokoll TCP ist (Schritt **608**). Wenn das Protokoll nicht TCP ist (also UDP), ruft das System einen Identifikator für einen Server-Netzknoten aus dem Eintrag in dem gewählten Speicherbereich der PDT ab (Schritt **612**). Andernfalls (wenn das Protokoll TCP ist) stellt das System fest, ob der gewählte Speicherbereich der PDT **304** eine Weiterleitungsliste aufweist (Schritt **609**). Wenn der gewählte Speicherbereich keine Weiterleitungsliste aufweist, ruft das System den Identifikator für den Server-Netzknoten aus dem Eintrag in dem gewählten Speicherbereich der PDT **304** ab (Schritt **612**). Wenn der gewählte Speicherbereich eine Weiterleitungsliste aufweist, stellt das System fest, ob die Client-IP-Adresse und Portnummer in einer Weiterleitungsliste auftreten (Schritt **610**). Durch die Weiterleitungsliste wird es möglich, dass bestehende TCP-Verbindungen weiter zu demselben Netzknoten gehen, wenn die PDT **304** verändert wird. Wenn dies dar Fall ist, ruft das System den Server-Identifikator aus der Weiterleitungsliste ab (Schritt **611**). Andernfalls ruft das System den Server-Identifikator aus dem gewählten Speicherbereich in der PDT **304** ab (Schritt **612**). Bei der bevorzugten Ausführungsform, bei der die Weiterleitungsliste und eine Kopie der PDT in einem Dienstobjekt geführt werden, müssen nur die Client-IP-Adresse und der Client-Port eines Eintrags in der Weiterleitungsliste mit der Client-IP-Adresse und dem Client-Port des Pakets verglichen werden, um zu bestimmen, ob eine Übereinstimmung besteht, da der vergleich des Pakets mit dem Dienstobjekt bereits die Dienst-IP-Adresse an den Dienstport angepasst hat.

[0066] Als nächstes leitet das System das Paket zu dem durch den Server-Identifikator angegebenen Server-Netzknoten weiter (Schritt **613**). Dieser Weiterleitungsprozess wird später mit Bezug auf **Fig.** 8 ausführlicher beschrieben.

[0067] Der Schnittstellennetzknoten ermöglicht es dem gewählten Server-Netzknoten dann, Rückkehrübermittlungen direkt zu dem Client zurückzusenden (Schritt **614**).

## Prozess des Nachschlagens eines Dienstes

[0068] **Fig.** 7 ist ein Flussdiagramm des Prozesses des Nachschlagens eines Dienstes für ein Paket gemäß einer Ausführungsform der vorliegenden Erfindung. Das System startet, indem es auf der Basis der Zieladresse in einer ersten Hash-Tabelle ein Nachschlagen durchführt (Schritt **702**). Bei der beschriebenen Ausführungsform basiert das Nachschlagen der Zieladresse in der ersten Hash-Tabelle auf einer Hashverarbeitung des Protokolls, der Dienst-IP-Adresse und der Portnummer des Dienstes. Wenn während dieses Nachschlagens ein Eintrag zurückgegeben wird, werden der zugeordnete Dienst, das zugeordnete Dienstobjekt und die zugeordnete Dienstgruppe zurückgegeben (Schritt **704**).

[0069] Andernfalls schlägt das System in einer zweiten Hash-Tabelle auf der Basis der Zieladresse einen skalierbaren Dienst nach (Schritt **706**). In diesem Fall werden nur das Protokoll und die IP-Adresse zur Durchführung des Nachschlagens benutzt. Der Grund dafür besteht darin, dass an dem zweiten Nachschlagen ein skalierbarer Dienst mit einer Eigenschaft der „Wildcard-Client-Affinität" beteiligt ist.

[0070] Wildcard-Client-Affinität versucht sicherzustellen, dass damit zusammenhängende Dienste auf demselben Server-Netzknoten für denselben Client für alle Dienst-Ports, einschließlich unregistrierter Ports, durchgeführt werden. Daher assoziiert die zweite Hash-Tabelle verwandte Dienste mit derselben IP-Adresse, aber mit verschiedenen Portnummern mit demselben Server-Netzknoten. Dies ist nützlich, wenn ein Netzknoten anfordert, dass der Client auf einem dynamisch zugewiesenen Port „zurückruft". Der Prozess ist abgeschlossen und es werden der zugeordnete Dienst, das zugeordnete Dienstobjekt und die zugeordnete Dienstgruppe zurückgegeben (Schritt **708**).

[0071] Wenn bei dem zweiten Nachschlagen kein Eintrag zurückgegeben wird, dann ist der Dienst ein nichtskalierbarer Dienst und das System signalisiert diese Tatsache (Schritt **710**), so dass Schritt **603** von **Fig.** 6 das Paket zu einem lokalen IP-Stapel sendet (Schritt **604**).

[0072] Bei einer Ausführungsform der vorliegenden Erfindung wählt das erste Nachschlagen mit Nicht-Affinität- und Client-Affinität-Lastausgleichsrichtlinientypen zu assoziierende Dienste und das zweite Nachschlagen wählt mit Wildcard-Client-Affinität-Lastausgleichsrichtlinientypen zu assoziierende Dienste, obwohl andere Anordnungen im Schutzumfang der Erfindung liegen.

## Prozess des Weiterleitens eines Pakets

[0073] **Fig.** 8 ist ein Flussdiagramm des Prozesses der Weiterleitung eines Pakets an einen Server gemäß einer Ausführungsform der vorliegenden Erfindung. Zu einem bestimmten Zeitpunkt während eines Initialisierungsprozesses stellt das System sicher, dass die IP-Adresse eines Dienstes auf der Loopback-Schnittstelle jedes Server-Netzknotens, der zur Durchführung des Dienstes verwendet wird, gehostet wird (Schritt **801**). Dadurch kann jeder Server-Netzknoten Pakete für den Dienst verarbeiten, obwohl der Dienst nicht auf einer öffentlichen Schnittstelle des Server-Netzknotens gehortet wird. Nachdem ein IP-Paket **400** empfangen worden ist und ein Dienst und ein Server-Netzknoten gewählt wurden (im Schritt **612** von **Fig.** 6), leitet das System das IP-Paket **400** aus dem Cluster-Vernetzungsmultiplexer **218** in dem Schnittstellennetzknoten an den IP-Stapel **216** in dem Server-Netzknoten **104** weiter. Dies umfasst das Konstruieren eines DLPI-Kopfteils **402**, einschließlich der MAC-Adresse des Server-Netzknotens **104** (Schritt **802**), und das anschließende Anfügen des DLPI-Kopfteils **402** an das IP-Paket **400** (siehe **Fig.** 4) (Schritt **804**).

[0074] Als nächstes sendet das System das IP-Paket **400** mit dem DLPI-Kopfteil **402** an die private Schnittstelle **224** in dem Schnittstellennetzknoten (Schritt **806**). Die private Schnittstelle **224** sendet das IP-Paket **400** mit DLPI-Kopfteil **402** an den Server-Netzknoten **104**. Der Server-Netzknoten **104** empfängt das IP-Paket **400** mit dem DLPI-Kopfteil **402** an der privaten Schnittstelle **226** (Schritt **808**). Als nächstes entfernt ein Treiber in dem Server-Netzknoten **104** den DLPJ-Kopfteil **402** von dem IP-Paket **400** (Schritt **810**). Das IP-Paket **400** wird dann in die unterste Position des IP-Stapels **216** auf dem Server-Netzknoten **104** eingespeist (Schritt **812**). Danach wird das IP-Paket **400** auf seinem Weg zu der Server-Instanz **203** durch den IP-Stapel **216** geleitet.

[0075] Man beachte, dass das herkömmliche Mittel der Verwendung eines Remote Procedure Call (RPC) oder eines Aufrufs der Schnittstellendefinitionssprache (IDL) zum Weiterleiten eines Pakets aus dem Schnittstellennetzknoten zu dem Server-Netzknoten **104** das Durchqueren eines IP-Stapels von einem RPC/IDL-Endpunkt zu der privaten Schnittstelle **224** in dem Schnittstellennetzknoten und das anschließende Durchqueren eines weiteren IP-Stapels wieder an dem Server-Netzknoten **104** von der privaten Schnittstelle **226** zu einem RPC/IDL-Endpunkt umfasst. Dies umfasst zwei IP-Stapeldurchquerungen und ist daher sehr ineffizient.

[0076] Im Gegensatz dazu beseitigt die in dem Flussdiagramm von **Fig.** 8 skizzierte Technik die zwei IP-Stapeldurchquerungen.

[0077] Man beachte außerdem, dass das System durch die Weiterleitung des Pakets zu dem Server-Netzknoten einen Lastausgleich zwischen mehreren redundanten wegen zwischen dem Schnittstellennetzknoten und dem Server-Netzknoten durchführen kann.

## Weiterleitungsliste

[0078] **Fig.** 10 zeigt ein verfahren zum Erzeugen oder Aktualisieren einer Weiterleitungsliste, wenn ein Server-Identifikator eines Speicherbereichs in einer PDT verändert wird. Solche Änderungen können aus verschiedenen Gründen vorgenommen werden. Ein Grund für das Ändern eines Server-Identifikators besteht darin, den Lastausgleich für eine PDT einer Dienstgruppe zu ändern. Ein Bediener kann den Lastausgleich zwischen Netzknoten als Mittel zum Abstimmen des Systems verändern.

[0079] Wenn ein Server-Identifikator eines Speicherbereichs einer PDT verändert wird, wird ein Server-Identifikator für einen alten Netzknoten mit einem Server-Identifikator eines neuen Netzknotens ersetzt (Schritt **1004**). Der Schnittstellennetzknoten prüft, ob der alte Netzknoten bestehende TCP-Verbindungen aufweist (Schritte **1005** und **1010**). Wenn es keine bestehenden TCP-Verbindungen gibt, ist der Prozess fertig (Schritt **1022**). Wenn der alte Netzknoten bestehende Verbindungen aufweist, wird abgefragt, ob der Speicherbereich eine Weiterleitungsliste aufweist (Schritt **1012**). Wenn der Speicherbereich keine Weiterleitungsliste aufweist, wird eine Weiterleitungsliste erzeugt (Schritt **1014**). Beide Zweige des Schritts **1012** fügen dann die bestehenden TCP-Verbindungen zu der Weiterleitungsliste hinzu (Schritt **1016**). Bei einer Ausführungsform werden alle bestehenden TCP-Verbindungen für Pakete mit derselben Dienst-IP-Adresse und demselben Dienstport wie die PDT zu der Weiterleitungsliste hinzugefügt. Bei einer anderen Ausführungsform werden nur die TCP-Verbindungen mit derselben Dienst-IP-Adresse und demselben Dienstport wie die PDT und die Kombinations-Quellen-IP-Adresse und der Quellenport, der in dem Speicherbereich im Hash-Verfahren verarbeitet werden kann, zu der Weiterleitungsliste hinzugefügt, so dass nur Verbindungen, die dem Speicherbereich entsprechen, zu der Weiterleitungsliste hinzugefügt werden. Der Vorteil des Ablegens aller Verbindungen in der Weiterleitungsliste besteht darin, dass dadurch durch das Hash-Verfahren, um zu sehen, welche Quel-

len-IP-Adresse und welcher Quellenport dem Speicherbereich entsprechen, erforderliche Zeit gespart wird. Der Vorteil, nur Verbindungen, die dem Speicherbereich entsprechen, zu der Weiterleitungsliste hinzuzufügen, besteht darin, dass dadurch die Weiterleitungslistengröße auf einem Minimum gehalten wird.

[0080] Wenn eine bestehende TCP-Verbindung beendet wird, sendet der alte Netzknoten eine Nachricht, die angibt, dass die Verbindung beendet wurde, und die Verbindung wird aus der Weiterleitungsliste gelöscht (Schritt **1018**). Wenn die Weiterleitungsliste leer wird, kann der Eintrag für die Weiterleitungsliste aus dem Speicherbereich entfernt werden (Schritt **1020**).

[0081] Die Benutzung der Weiterleitungsliste wird oben in den Schritten **608** bis **613** von **Fig.** 6 beschrieben.

Beispiel

[0082] Zum Beispiel kann in dem in **Fig.** 1 und **Fig.** 2 gezeigten System, wenn die ursprünglichen Lastausgleichsgewichte für eine Dienstgruppe 50% für Netzknoten und 50% für Netzknoten **102** betrugen und der Bediener findet, dass der Netzknoten überlastet ist, der Bediener die Lastausgleichsgewichte auf 25% für den Netzknoten und 75% für den Netzknoten **102** ändern.

[0083] Eine Vorher-PDT für ein Master-Dienstobjekt {TCP, www.sun.com, 80} kann wie nachfolgend gezeigt aussehen:

Vorher-PDT

| Speicherbereich | Netzknoten | Weiterleitungsliste |
|---|---|---|
| 1 | 103 | keine |
| 2 | 103 | keine |
| 3 | 102 | keine |
| 4 | 102 | keine |

[0084] Vor der Änderung kann ein erstes Paket, das ein typisches IP-Protokoll verwendet, während einer Session einer ersten Verbindung an dem Schnittstellennetzknoten ankommen (Schritt **1004**). Das IP-Paket besitzt einen 5-Tupel-Kopfteil in der Form {Protokoll, Dienst-IP-Adresse, Dienstport, Quellen-IP-Adresse, Quellenport}. Zum Beispiel kann das erste Paket das 5-Tupel {TCP, www.sun.com, 80, ibm.com, 84} aufweisen. Es zeigt sich, dass das Dienstobjekt mit der Vorher-PDT mit dem Paket übereinstimmt (Schritt **602**). In diesem Beispiel ist der Lastausgleichsrichtlinientyp gewichtet. Deshalb werden die Quellen-IP-Adresse und der Quellenport über der PDT hash-verarbeitet (Schritt **607**). Es gibt verschiedene Verfahren zur Hash-Verarbeitung der Quellenadresse und des Quellenports des Pakets über der PCT. Eines besteht darin, den Modul der Summe der Quellenadresse und des Quellenports zu nehmen. Bei einem Beispiel für eine-Art dies durchzuführen ist der Divisor, wenn ibm.com eine IP-Adresse besitzt, die im Hash-Verfahren 98.942 ergibt, da nur vier Speicherbereiche in PDT vorliegen, für den Modul darin 4. Deshalb lautet der verwendete Speicherbereich:

(IP-Quellenadresse + Quellenport) Mod Anzahl von Speicherbereichen = (98.942+84) Mod 4 = 2.

[0085] Da das Paket ein TCP-Protokoll aufweist (Schritt **608**), wird die Vorher-PDT untersucht, um zu sehen, ob Speicherbereich 2 eine Weiterleitungsliste aufweist (Schritt **609**). Da Speicherbereich 2 keine Weiterleitungsliste aufweist, wird das Paket zu dem im Speicherbereich 2 aufgelisteten Netzknoten weitergeleitet (Schritte **612** und **613**), d. h. Netzknoten **103**. Während dieser Verbindung ist es also wünschenswert, alle Pakete in der Verbindung zu dem Netzknoten zu lenken.

[0086] Als Folge wechselt bei der Änderung der Lastausgleichsgewichte von 50% für den Netzknoten **103** und 50% für den Netzknoten **102** auf 25% für den Netzknoten **103** und 75% für den Netzknoten **102** die PDT in dem Dienstobjekt zu der unten als Nachher-PDT gezeigten Konfiguration:

[0087] Der alte Netzknoten (Netzknoten **103**) wird auf etwaige bestehende Verbindungen abgefragt (Schritt **1008**). Da die Verbindung mit „ibm.com, 84" immer noch besteht (Schritt **1010**), wird Speicherbereich 2 geprüft, um zu sehen, ob Speicherbereich 2 eine Weiterleitungsliste aufweist. Da Speicherbereich 2 keine Weiterleitungsliste aufweist (Schritt **1012**), wird die Weiterleitungsliste 1. erzeugt (**1014**) und die bestehenden Verbindungen werden wie nachfolgend gezeigt zu der Weiterleitungsliste 1 hinzugefügt (Schritt **1016**):

Nachher-PDT

| Speicherbereich | Netzknoten | Weiterleitungsliste |
|---|---|---|
| 1 | 103 | keine |
| 2 | 102 | Weiterleitungsliste 1 |
| 3 | 102 | keine |
| 4 | 102 | keine |

Weiterleitungsliste 1

| Quellenadresse | alter Knoten |
|---|---|
| ibm.com, 84 | 103 |

[0088] In diesem Beispiel werden nur die Quellenadresse und der Port in der Weiterleitungsliste gespeichert. Bei anderen Ausführungsformen können auch das Protokoll, die Dienstadresse und der Port in der Weiterleitungsliste gespeichert werden.
[0089] Ein zweites Paket während derselben Session und Verbindung wie das erste Paket würde das 5-Tupel {TCP, www.sun.com, 80, ibm.com, 84} aufweisen. Es zeigt sich, dass dasselbe Dienstobjekt, das nun die Nachher-PDT aufweist, mit dem zweiten Paket übereinstimmt (Schritt **602**). In diesem Fall ist der Lastausgleichsrichtlinientyp gewichtet. Deshalb werden die Quellen-IP-Adresse und der Quellenport über der PDT hash-verarbeitet (Schritt **607**). Der benutzte Speicherbereich lautet deshalb:

(IP-Quellenadresse + Quellenport) Mod Anzahl von Speicherbereichen = (98.942+84) Mod 4 = 2.

[0090] Da das Paket ein TCP-Protokoll aufweist (Schritt **608**), wird die Nachher-PDT untersucht, um zu sehen, ob Speicherbereich 2 eine Weiterleitungsliste aufweist (Schritt **609**). Da der Speicherbereich 2 eine Weiterleitungsliste (Weiterleitungsliste 1) aufweist (Schritt **609**), wird die Weiterleitungsliste 1 durchsucht, um zu sehen, ob es etwaige Übereinstimmungen mit der Quellen-IP-Adresse und dem Quellen-Port des zweiten Pakets gibt (Schritt **610**). Der einzige Eintrag in der Weiterleitungsliste 1 erweist sich als Übereinstimmung mit dem zweiten Paket, da beide die Quellenadresse {ibm.com, 84} referenzieren, und die Server-ID zu dem alten Netzknoten wird aus der Weiterleitungsliste 1 abgerufen (Schritt **611**). Als Folge leitet der Schnittstellennetzknoten das Paket zu dem in der Weiterleitungsliste aufgelisteten alten Netzknoten weiter, d. h. Netzknoten **103** (Schritt **613**). Obwohl der Lastausgleich verändert wurde, wodurch die PDT verändert wurde, gingen deshalb TCP-Pakete derselben Verbindung weiter zu denselben Netzknoten, wodurch die Unterbrechung der Verbindung vermieden wird.
[0091] Wenn eine zweite Verbindung nach der Änderung gestartet wird, während die erste Verbindung von demselben Client fortgesetzt wird, verwendet der Client einen anderen Client-Port für die zweite Verbindung. Ein drittes Paket, das als Teil dieser zweiten Verbindung gesendet wird, kann ein 5-Tupel aufweisen, das zum Beispiel {TCP, www.sun.com, 80, ibm.com, 84} lauten kann. Es zeigt sich, dass das Dienstobjekt mit der Nachher-PDT mit dem dritten Paket übereinstimmt (Schritt **602**). In diesem Beispiel ist der Lastausgleichrichtlinientyp gewichtet. Deshalb werden die Quellen-IP-Adresse und der Quellenport über der PDP hashverarbeitet (Schritt **607**). Der verwendete Speicherbereich lautet deshalb:

(IP-Quellenadresse + Quellenport) Mod Anzahl von Speicherbereichen = (98.942+80) Mod 4 = 2.

[0092] Da das dritte Paket ein TCP-Protokoll aufweist (Schritt **608**), wird die Nachher-PDT untersucht, um zu sehen, ob der Speicherbereich 2 eine Weiterleitungsliste aufweist (Schritt **609**). Da der Speicherbereich 2 eine Weiterleitungsliste (Weiterleitungsliste 1) aufweist (Schritt **609**), wird die Weiterleitungsliste 1 durchsucht, um zu sehen, ob es etwaige Übereinstimmungen mit der Quellen-IP-Adresse und dem Quellenport des zweiten Pakets gibt (Schritt **610**). Da das dritte Paket einen anderen Quellenport als der einzige Eintrag in der Weiterleitungsliste aufweist, wird keine Übereinstimmung gefunden. Als Ergebnis leitet der Schnittstellennetzknoten das Paket zu dem in dem Speicherbereich 2 der Nachher-PDT aufgelisteten neuen Netzknoten weiter, d. h. den Netzknoten **102** (Schritte **612** und **613**). Neue Verbindungen, die nicht mit Einträgen in der Weiterleitungsliste übereinstimmen, werden deshalb gemäß der aktualisierten PDT weitergeleitet.
[0093] Nachdem die erste Verbindung beendet wurde, wird der Eintrag aus der Weiterleitungsliste 1 entfernt (Schritt **1018**). Da die Weiterleitungsliste 1 leer ist, wird die Weiterleitungsliste 1 aus dem Speicherbereich 2 entfernt (Schritt **1020**). Da die Weiterleitungsliste auf dem Speicherbereich 2 entfernt ist, wird jedes an den Speicherbereich 2 Berichtete Paket zu dem Netzknoten **102** weitergeleitet.

[0094] Obwohl die vorliegende Erfindung anhand mehrerer bevorzugter Ausführungsformen beschrieben wurde, gibt es Veränderungen, Permutationen und Äquivalente, die in den Schutzumfang der vorliegenden Erfindung fallen. Außerdem sollte beachtet werden, dass es viele alternative Arten der Implementierung der Verfahren und Vorrichtungen der vorliegenden Erfindung gibt.

## Patentansprüche

1. Verfahren (**600**) zur Verteilung von Datenpaketen an Server-Netzknoten (**102**–**104**) in einem Netzknoten-Cluster (**100**), das die folgenden Schritte umfasst:
– Empfangen (**601**) eines Pakets (**400**) an einem Schnittstellennetzknoten (**103**) im Netzknoten-Cluster, wobei das Paket eine Zieladresse (**404**) und eine Quelladresse (**406**) einschließt;
– Anpassen (**602**) des Pakets an ein Dienstobjekt (**308**), das einem Dienst zugeordnet ist, unter Verwendung der Zieladresse, wobei der Dienst durch die Zieladresse bestimmt wird;
– Abbilden der Quelladresse in einem Speicherbereich einer Mehrzahl von Speicherbereichen in einer Paketverteilungsliste (**304**), wobei jeder Speicherbereich einen Identifikator für einen der Server-Netzknoten im Netzknoten-Cluster enthält, wobei der Speicherbereich dem an das Paket angepassten Dienstobjekt zugeordnet ist;
– Feststellen (**609**), ob dem Speicherbereich eine Weiterleitungsliste zugeordnet ist;
– wenn festgestellt wurde, dass dem Speicherbereich eine Weiterleitungsliste zugeordnet ist, Feststellen (**610**), ob die Quelladresse (**406**) mit einem Eintrag in der Weiterleitungsliste übereinstimmt;
– wenn es eine Übereinstimmung mit einem Eintrag in der Weiterleitungsliste gibt, Senden (**613**) des Pakets an den Server-Netzknoten, auf den durch die Übereinstimmung mit dem Eintrag in der Weiterleitungsliste verwiesen wird, wodurch sichergestellt wird, dass eine bereits bestehende Verbindung nicht unterbrochen wird; und
– wenn es keine Übereinstimmung mit einem Eintrag in der Weiterleitungsliste gibt, Senden (**613**) des Pakets an den Server-Netzknoten, auf den durch den Speicherbereich verwiesen wird, in dem die Quelladresse des Pakets abgebildet ist.

2. Verfahren nach Anspruch 1, bei dem der Schritt zur Feststellung (**610**), ob die Quelladresse mit einem Eintrag in der Weiterleitungsliste übereinstimmt, die folgenden Schritte umfasst:
– Feststellen, ob es einen Eintrag in der Weiterleitungsliste in dem Speicherbereich gibt, in dem die Quelladresse (**406**) des Pakets (**400**) abgebildet ist;
– wenn es keinen Eintrag in der Weiterleitungsliste in dem Speicherbereich gibt, in dem die Quelladresse des Pakets abgebildet ist, Feststellen, dass es keine Übereinstimmung mit einem Eintrag in der Weiterleitungsliste gibt; und
– wenn es einen Eintrag in dem einen Speicherbereich gibt, in dem die Quelladresse des Pakets abgebildet ist, Durchsuchen (**611**) der Weiterleitungsliste nach einer Übereinstimmung mit der Quelladresse des Pakets.

3. Verfahren nach Anspruch 2, welches außerdem die folgenden Schritte umfasst:
– Aktualisieren (**1004**) des Speicherbereichs in der Paketverteilungsliste (**304**), welche dem Dienstobjekt (**308**) zugeordnet ist, indem der Identifikator im Speicherbereich von einem alten Netzknoten auf einen neuen Netzknoten geändert wird; und
– Aktualisieren einer Weiterleitungsliste, die dem aktualisierten Speicherbereich zugeordnet ist.

4. verfahren nach Anspruch 3, bei dem der Schritt zur Aktualisierung der Weiterleitungsliste die folgenden Schritte umfasst:
– Abfragen (**1008**) des alten Netzknotens nach bestehenden Verbindungen; und
– Hinzufügen (**1016**) der bestehenden Verbindungen des alten Netzknotens zur Weiterleitungsliste.

5. Verfahren nach Anspruch 4, bei dem die bestehenden Verbindungen Pakete mit einer gemeinsamen Zieladresse (**404**) und Quelladresse (**406**) aufweisen und bei dem der Schritt zum Hinzufügen der bestehenden Verbindungen des alten Netzknotens zur Weiterleitungsliste die folgenden Schritte umfasst:
– Durchsuchen der bestehenden Verbindungen des alten Netzknotens nach bestehenden Verbindungen mit Zieladressen, die mit dem Dienst übereinstimmen, welcher dem Dienstobjekt zugeordnet ist; und
– Hinzufügen zur Weiterleitungsliste (**1016**) der gefundenen bestehenden Verbindungen des alten Netzknotens mit einer Zieladresse, welche mit dem Dienst übereinstimmt, der dem Dienstobjekt zugeordnet ist.

6. verfahren nach Anspruch 5, welches außerdem den Schritt umfasst, bei dem eine beendete Verbindung der bestehenden Verbindungen auf der Weiterleitungsliste gelöscht (**1018**) wird.

7. Verfahren nach Anspruch 1, welches außerdem die folgenden Schritte umfasst:
– Prüfen (**608**) des Protokolls des Pakets (**400**); und
– Senden (**613**) des Pakets an den durch den Speicherbereich identifizierten Server-Netzknoten, wenn das Protokoll kein TCP-Protokoll ist, wobei die Schritte zum Überprüfen des Protokolls (**608**) und zum Senden des Pakets (**613**) zeitlich vor dem Schritt zur Feststellung (**610**), ob die Quelladresse (**406**) mit einem Eintrag übereinstimmt, liegen.

8. Verfahren nach Anspruch 7, bei dem die Quelladresse (**406**) eine Client IP-Adresse und eine Client Port-Adresse umfasst.

9. verfahren nach Anspruch 4, bei dem die bestehenden Verbindungen Pakete mit einer gemeinsamen Zieladresse (**404**) und Quelladresse (**406**) aufweisen und bei dem der Schritt zum Hinzufügen (**1016**) der bestehenden Verbindungen des alten Netzknotens zur Weiterleitungsliste die folgenden Schritte umfasst:
– Durchsuchen der bestehenden Verbindungen des alten Netzknotens nach Verbindungen mit Zieladressen, die mit dem Dienst übereinstimmen, der dem Dienstobjekt zugeordnet ist; und
– Hinzufügen zur Weiterleitungsliste der bestehenden Verbindungen des alten Netzknotens mit einer Zieladresse, die mit dem Dienst übereinstimmt, welcher dem Dienstobjekt zugeordnet ist, und mit einer Quelladresse, die in dem Speicherbereich abgebildet ist, welcher der Weiterleitungsliste zugeordnet ist.

10. Verfahren nach Anspruch 9, welches außerdem den Schritt umfasst, bei dem ein Eintrag einer beendeten Verbindung der bestehenden Verbindungen auf der weiterleitungsliste gelöscht (**1018**) wird.

11. Verfahren nach Anspruch 10, welches außerdem die folgenden Schritte umfasst:
– Überprüfen (**608**) des Protokolls des Pakets; und
– Senden (**613**) des Pakets an den durch den Speicherbereich identifizierten Server-Netzknoten, wenn das Protokoll kein TCP-Protokoll ist, wobei die Schritte zum Überprüfen des Protokolls und zum Senden des Pakets zeitlich vor dem Schritt zur Feststellung (**610**), ob die Quelladresse (**406**) mit einem Eintrag übereinstimmt, liegen.

12. Verfahren nach Anspruch 11, bei dem die Quelladresse (**406**) eine Client IP-Adresse und eine Client Port-Adresse umfasst.

13. Computerlesbares Speichermedium, das Befehle speichert, die, wenn diese von einem Computer ausgeführt werden, den Computer dazu veranlassen, ein Verfahren zur Verteilung von Paketen an Server-Netzknoten in einem Netzknoten-Cluster auszuführen, welches die folgenden Schritte umfasst:
– Empfangen (**601**) eines Pakets (**400**) an einem Schnittstellennetzknoten (**103**) im Netzknoten-Cluster (**100**), wobei das Paket eine Zieladresse (**404**) und eine Quelladresse (**406**) einschließt;
– Anpassen (**602**) des Pakets an ein Dienstobjekt (**308**), das einem Dienst zugeordnet ist, unter Verwendung der Zieladresse, wobei der Dienst durch die Zieladresse bestimmt wird;
– Abbilden der Quelladresse in einem Speicherbereich einer Mehrzahl von Speicherbereichen in einer Paketverteilungsliste (**304**), wobei jeder Speicherbereich einen Identifikator für einen der Server-Netzknoten im Netzknoten-Cluster enthält, wobei der Speicherbereich dem an das Paket angepassten Dienstobjekt zugeordnet ist;
– Feststellen (**609**), ob dem Speicherbereich eine Weiterleitungsliste zugeordnet ist;
– wenn festgestellt wurde, dass dem Speicherbereich eine Weiterleitungsliste zugeordnet ist, Feststellen (**610**), ob die Quelladresse (**406**) mit einem Eintrag in der Weiterleitungsliste übereinstimmt;
– wenn es eine Übereinstimmung mit einem Eintrag in der Weiterleitungsliste gibt, Senden (**613**) des Pakets an den Server-Netzknoten, auf den durch die Übereinstimmung verwiesen wird; und
– wenn es keine Übereinstimmung mit einem Eintrag in der Weiterleitungsliste gibt, Senden (**613**) des Pakets an den Server-Netzknoten, auf den durch den Speicherbereich verwiesen wird, in dem die Quelladresse des Pakets abgebildet ist.

14. Computerlesbares Speichermedium nach Anspruch 13, bei dem der Schritt zur Feststellung (**610**), ob die Quelladresse (**406**) mit einem Eintrag in der Weiterleitungsliste übereinstimmt, die folgenden Schritte umfasst:
– Feststellen, ob es einen Eintrag in der Weiterleitungsliste in dem Speicherbereich gibt, in dem die Quelladresse des Pakets (**400**) abgebildet ist;
– wenn es keinen Eintrag in der Weiterleitungsliste in dem Speicherbereich gibt, in dem die Quelladresse des Pakets abgebildet ist, Feststellen, dass es keine Übereinstimmung mit einem Eintrag in dar Weiterleitungsliste gibt; und

– wenn es einen Eintrag in dem Speicherbereich gibt, in dem die Quelladresse des Pakets abgebildet ist, Durchsuchen (**611**) der Weiterleitungsliste nach einer Übereinstimmung mit der Quelladresse des Pakets.

15. Computerlesbares Speichermedium nach Anspruch 14, das außerdem die folgenden Schritte umfasst:
– Aktualisieren (**1004**) des Speicherbereichs in der Paketverteilungsliste (**304**), welche dem Dienstobjekt (**308**) zugeordnet ist, indem der Identifikator im Speicherbereich von einem alten Netzknoten auf einen neuen Netzknoten geändert wird; und
– Aktualisieren einer Weiterleitungsliste, die dem aktualisierten Speicherbereich zugeordnet ist, wobei der Schritt zur Aktualisierung der Weiterleitungsliste die folgenden Schritte umfasst:
– Abfragen (**1008**) des alten Netzknotens nach bestehenden Verbindungen; und
– Hinzufügen (**1016**) der bestehenden Verbindungen des alten Netzknotens zur Weiterleitungsliste.

16. Computerlesbares Speichermedium nach Anspruch 15, bei dem die bestehenden Verbindungen Pakete mit einer gemeinsamen Zieladresse (**404**) und Quelladresse (**406**) aufweisen und bei dem der Schritt zum Hinzufügen der bestehenden Verbindungen des alten Netzknotens zur Weiterleitungsliste die folgenden Schritte umfasst:
– Durchsuchen der bestehenden Verbindungen des alten Netzknotens nach bestehenden Verbindungen mit Zieladressen, die mit dem Dienst übereinstimmen, welcher dem Dienstobjekt zugeordnet ist;
– Hinzufügen zur Weiterleitungsliste (**1016**) der gefundenen bestehenden Verbindungen des alten Netzknotens mit einer Zieladresse, welche mit dem Dienst übereinstimmt, der dem Dienstobjekt zugeordnet ist;
– Löschen (**1018**) aus der Weiterleitungsliste eines Eintrags einer beendeten Verbindung der bestehenden Verbindungen;
– Prüfen (**608**) des Protokolls des Pakets; und
– Senden (**613**) des Pakets an den durch den Speicherbereich identifizierten Server-Netzknoten, wenn das Protokoll kein TCP-Protokoll ist, wobei die Schritte zum Überprüfen des Protokolls und zum Senden des Pakets zeitlich vor dem Schritt zur Feststellung, ob die Quelladresse mit einem Eintrag übereinstimmt, liegen.

17. Computerlesbares Speichermedium nach Anspruch 15, bei dem die bestehenden Verbindungen Pakete mit einer gemeinsamen Zieladresse (**404**) und Quelladresse (**406**) aufweisen und bei dem der Schritt zum Hinzufügen der bestehenden Verbindungen des alten Netzknotens zur Weiterleitungsliste die folgenden Schritte umfasst:
– Durchsuchen der bestehenden Verbindungen des alten Netzknotens nach Verbindungen mit Zieladressen, die mit dem Dienst des Dienstobjekts übereinstimmen;
– Hinzufügen zur Weiterleitungsliste (**1016**) der bestehenden Verbindungen des alten Netzknotens mit einer Zieladresse, die mit dem Dienst des Dienstobjekts übereinstimmt, und mit einer Quelladresse, die in dem Speicherbereich der Weiterleitungsliste abgebildet ist; und wobei dieser außerdem die folgenden Schritte umfasst:
– Löschen (**1018**) aus dar Weiterleitungsliste eines Eintrags einer beendeten Verbindung der bestehenden Verbindungen;
– Prüfen (**608**) des Protokolls des Pakets; und
– Senden (**613**) des Pakets an den durch den Speicherbereich identifizierten Server-Netzknoten, wenn das Protokoll kein TCP-Protokoll ist, wobei die Schritte zum Überprüfen des Protokolls und zum Senden des Pakets zeitlich vor dem Schritt zur Feststellung, ob die Quelladresse mit einem Eintrag übereinstimmt, liegen.

18. Gerät zum Verteilen von Paketen an Server-Netzknoten (**102**–**104**) in einem Netzknoten-Cluster (**100**), bei dem einer der Server-Netzknoten ein Schnittstellennetzknoten (**103**) ist, welches umfasst:
– einen Empfangsmechanismus (**221**) im Schnittstellennetzknoten, der für den Empfang eines Pakets (**400**) am Schnittstellennetzknoten im Netzknoten-Cluster konfiguriert ist, wobei das Paket eine Zieladresse (**404**) und eine Quelladresse (**406**) einschließt;
– einen Mechanismus zum Anpassen des Dienstobjekts (**308**) im Schnittstellennetzknoten, der so konfiguriert ist, dass er das Paket an ein Dienstobjekt anpasst, welches einem Dienst zugeordnet ist, unter Verwendung der Zieladresse des Pakets, und wobei der Dienst durch die Zieladresse bestimmt wird;
– einen Abbildungsmechanismus im Schnittstellennetzknoten, der so konfiguriert ist, dass er die Quelladresse in einem Speicherbereich einer Mehrzahl von Speicherbereichen in einer Paketverteilungsliste (**304**) abbildet, wobei jeder Speicherbereich einen Identifikator für einen der Server-Netzknoten im Netzknoten-Cluster enthält und wobei der Speicherbereich dem an das Paket angepassten Dienstobjekt zugeordnet ist;
– einen Feststellungsmechanismus, der so konfiguriert ist, dass er feststellt, ob eine Weiterleitungsliste einem Speicherbereich zugeordnet ist;
– einen Feststellungsmechanismus im Schnittstellennetzknoten, der so konfiguriert ist, dass er feststellt, ob die Quelladresse mit einem Eintrag in der Weiterleitungsliste übereinstimmt;

– einen Sendemechanismus im Schnittstellennetzknoten bei Übereinstimmung, der so konfiguriert ist, dass er das Paket an den Server-Netzknoten sendet, auf den durch die Übereinstimmung verwiesen wird, wenn es eine Übereinstimmung mit einem Eintrag in der weiterleitungsliste gibt;
– eine Sendemechanismus im Schnittstellennetzknoten bei Nicht-Übereinstimmung, der so konfiguriert ist, dass er das Paket an den Server-Netzknoten sendet, der durch den Speicherbereich, in dem die Quelladresse des Pakets abgebildet ist, identifiziert wird.

19. Gerät nach Anspruch 18, bei dem der Feststellungsmechanismus umfasst:
– einen Feststellungsmechanismus für die Weiterleitungsliste, der so konfiguriert ist, dass er feststellt, ob es einen Eintrag in der Weiterleitungsliste in dem Speicherbereich gibt, in dem die Quelladresse (**406**) des Pakets (**400**) abgebildet ist;
– einen Verweismechanismus bei Nicht-Übereinstimmung, der so konfiguriert ist, dass er feststellt, dass es keine Übereinstimmung mit einem Eintrag in der Weiterleitungsliste gibt, wenn es keinen Eintrag in der Weiterleitungsliste in dem Speicherbereich gibt, in dem die Quelladresse des Pakets abgebildet ist; und
– einen Mechanismus zum Durchsuchen der Weiterleitungsliste, der so konfiguriert ist, dass er die Weiterleitungsliste nach einer Übereinstimmung mit der Quelladresse des Pakets durchsucht, wenn es einen Eintrag in der Weiterleitungsliste in dem Speicherbereich gibt, in dem die Quelladresse des Pakets abgebildet ist.

20. Gerät nach Anspruch 19, das außerdem umfasst:
– einen Mechanismus zur Aktualisierung des Speicherbereichs, der so konfiguriert ist, dass er den Speicherbereich in der Paketverteilungsliste (**304**) des Dienstobjekts (**308**) aktualisiert, indem der Identifikator im Speicherbereich von einem alten Netzknoten auf einen neuen Netzknoten geändert wird; und
– einen Mechanismus zur Aktualisierung der Weiterleitungsliste, der so konfiguriert ist, dass er eine Weiterleitungsliste aktualisiert, die dem aktualisierten Speicherbereich zugeordnet ist, wobei der Mechanismus zur Aktualisierung der Weiterleitungsliste umfasst:
– einen Abfragemechanismus, der so konfiguriert ist, dass er den alten Netzknoten nach bestehenden Verbindungen abfragt; und
– einen Hinzufügungsmechanismus, der so konfiguriert ist, dass er bestehende Verbindungen des alten Netzknotens der Weiterleitungsliste als Einträge hinzufügt.

21. Gerät nach Anspruch 20, bei dem die bestehenden Verbindungen Pakete mit einer gemeinsamen Zieladresse (**404**) und Quelladresse (**406**) aufweisen, und wobei der Hinzufügungsmechanismus umfasst:
– einen Suchmechanismus, der so konfiguriert ist, dass er den alten Netzknoten nach bestehenden Verbindungen mit Zieladressen durchsucht, welche mit dem Dienst übereinstimmen, der dem Dienstobjekt zugeordnet ist;
– einen Mechanismus zum Hinzufügen gefundener bestehender Verbindungen, der so konfiguriert ist, dass er gefundene bestehende Verbindungen des alten Netzknotens mit einer Zieladresse, welche mit dem Dienst übereinstimmt, der dem Dienstobjekt zugeordnet ist, zur Weiterleitungsliste hinzufügt; und wobei das Gerät außerdem umfasst:
– einen Löschmechanismus, der so konfiguriert ist, dass er einen Eintrag einer beendeten Verbindung der bestehenden Verbindungen aus der Weiterleitungsliste löscht;
– einen Mechanismus zur Überprüfung des Protokolls, der so konfiguriert ist, dass er das Protokoll des Pakets prüft; und
– einen Sendemechanismus bei einem Nicht-TCP-Protokoll, der so konfiguriert ist, dass er das Paket an den durch den Speicherbereich identifizierten Server-Netzknoten schickt, wenn das Protokoll kein TCP-Protokoll ist.

22. Gerät nach Anspruch 20, bei dem die bestehenden Verbindungen Pakete aufweisen, die eine gemeinsame Zieladresse (**404**) und Quelladresse (**406**) haben, und wobei der Hinzufügungsmechanismus umfasst:
– einen Suchmechanismus, der so konfiguriert ist, dass er die bestehenden Verbindungen des alten Netzknotens mit Zieladressen durchsucht, welche mit dem Dienst übereinstimmen, der dem Dienstobjekt zugeordnet ist;
– einen Mechanismus zum Hinzufügen gefundener bestehender Verbindungen zur Weiterleitungsliste, der so konfiguriert ist, dass er die bestehenden Verbindungen des alten Netzknotens mit einer Zieladresse, welche mit dem Dienst übereinstimmt, der dem Dienstobjekt zugeordnet ist, und einer Quelladresse, welche in dem Speicherbereich der Weiterleitungsliste abgebildet ist; und wobei das Gerät außerdem umfasst:
– einen Löschmechanismus zum Löschen aus der Weiterleitungsliste eines Eintrags einer beendeten Verbindung der bestehenden Verbindungen;
– einen Mechanismus zum Überprüfen des Protokolls, der so konfiguriert ist, dass er das Protokoll des Pakets prüft; und

– einen Sendemechanismus bei einem Nicht-TCP-Protokoll, der so konfiguriert ist, dass er das Paket an den durch den Speicherbereich identifizierten Server-Netzknoten schickt, wenn das Protokoll kein TCP-Protokoll ist.

Es folgen 8 Blatt Zeichnungen

**Anhängende Zeichnungen**

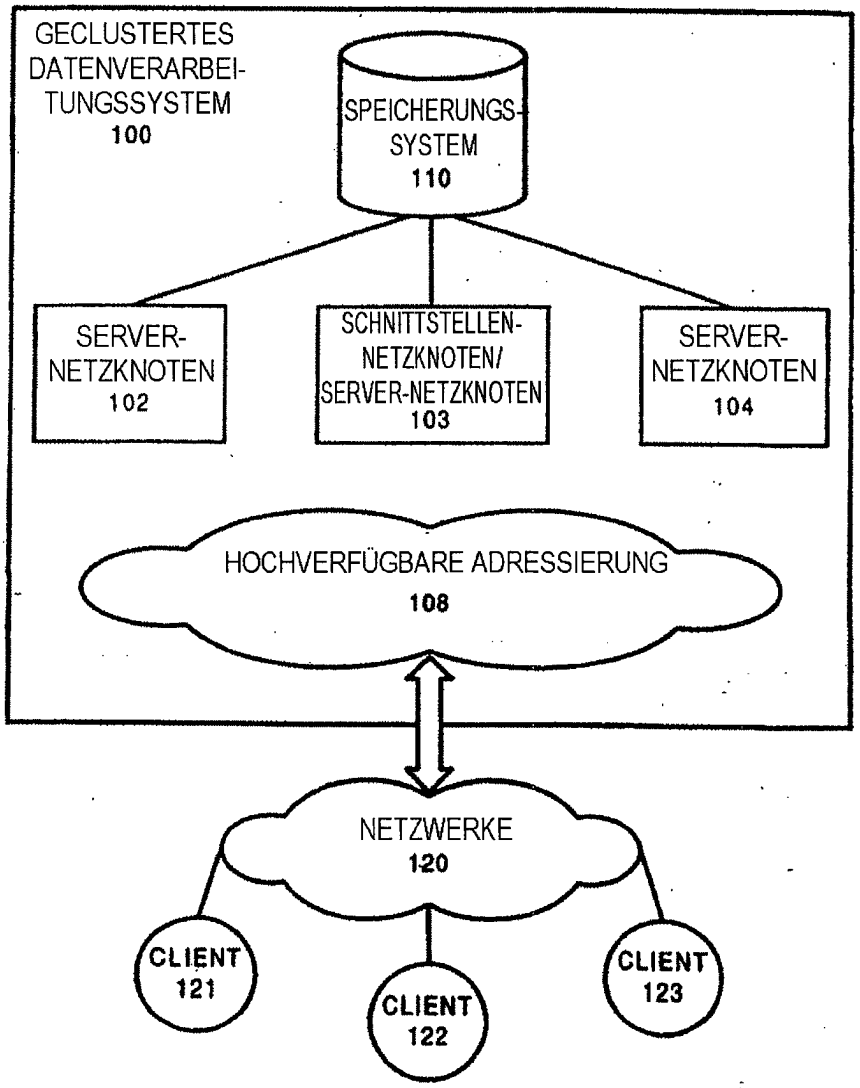


**FIG. 1**

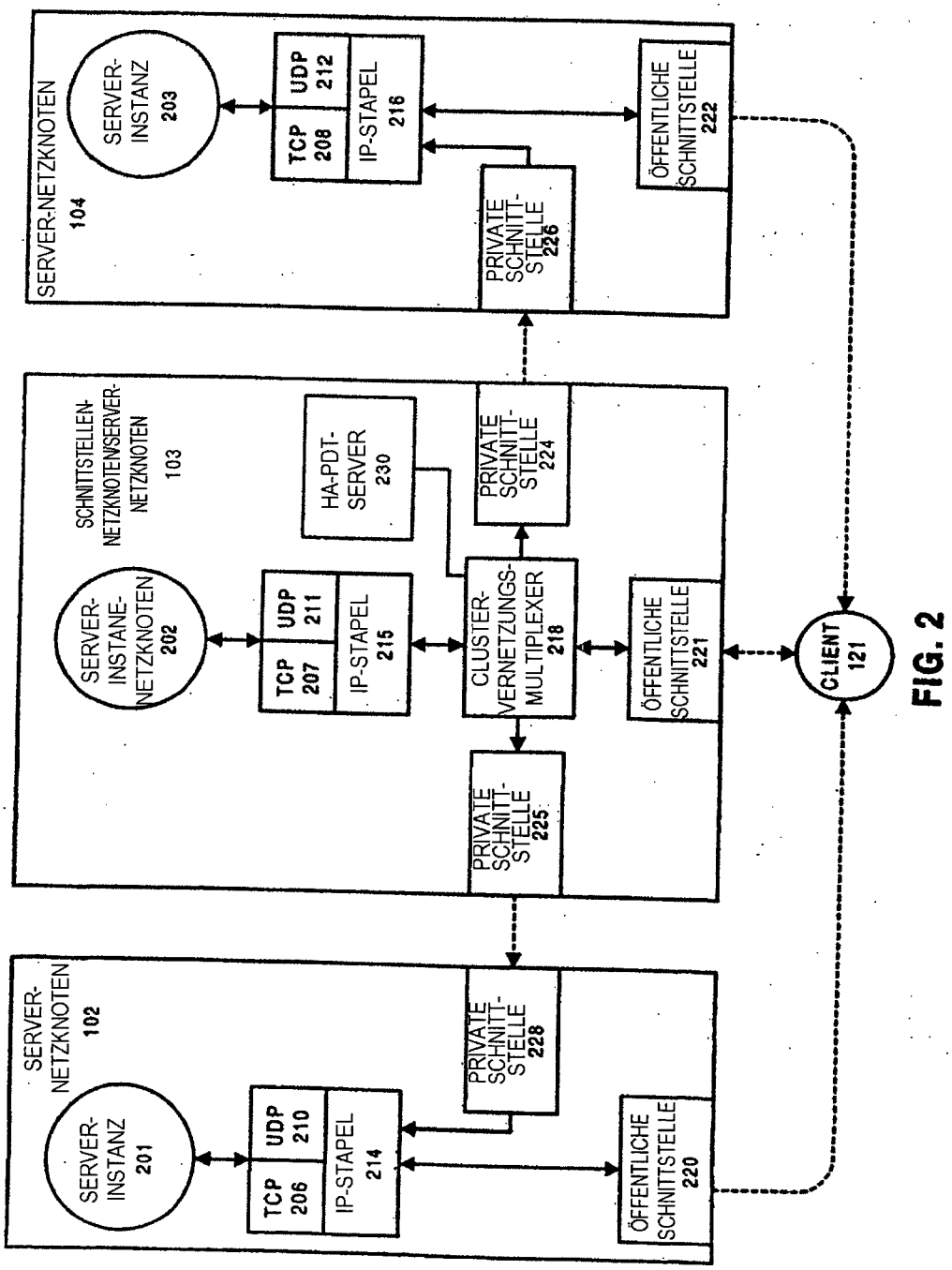
SERVER-NETZKNOTEN
104
SERVER-INSTANZ
203
TCP
208
UDP
212
IP-STAPEL
216
PRIVATE SCHNITT-STELLE
226
ÖFFENTLICHE SCHNITTSTELLE
222
SCHNITTSTELLEN-NETZKNOTEN/SERVER-NETZKNOTEN
103
HA-PDT-SERVER
230
PRIVATE SCHNITT-STELLE
224
SERVER-INSTANE-NETZKNOTEN
202
TCP
207
UDP
211
IP-STAPEL
215
CLUSTER-VERNETZUNGS-MULTIPLEXER
218
ÖFFENTLICHE SCHNITTSTELLE
221
CLIENT
121
PRIVATE SCHNITT-STELLE
225
SERVER-NETZKNOTEN
102
PRIVATE SCHNITT-STELLE
228
SERVER-INSTANZ
201
TCP
206
UDP
210
IP-STAPEL
214
ÖFFENTLICHE SCHNITTSTELLE
220

FIG. 2

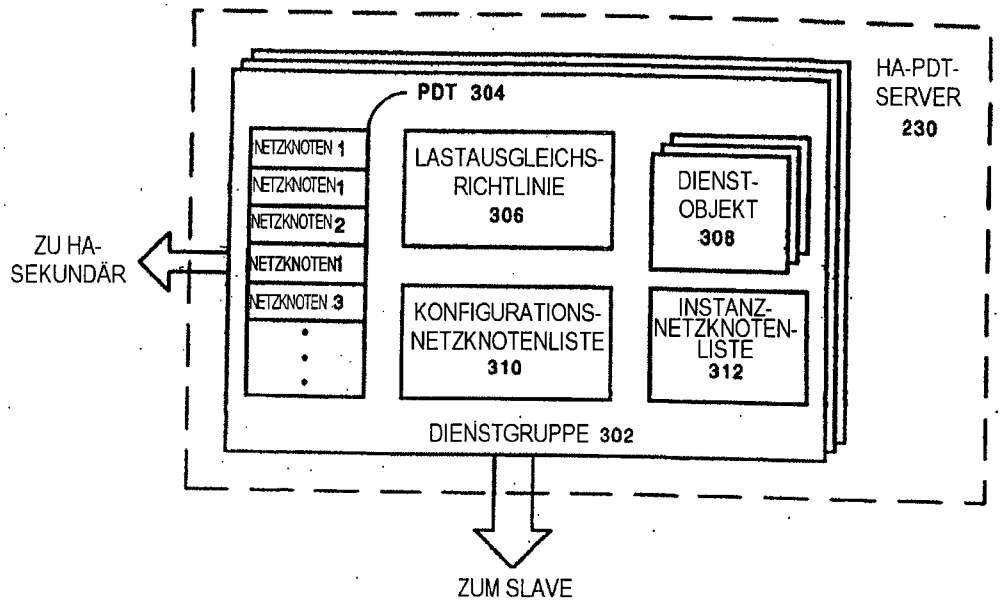
HA-PDT-SERVER
230
PDT 304
NETZKNOTEN 1
NETZKNOTEN 1
NETZKNOTEN 2
NETZKNOTEN 1
NETZKNOTEN 3
LASTAUSGLEICHS-RICHTLINIE
306
DIENST-OBJEKT
308
KONFIGURATIONS-NETZKNOTENLISTE
310
INSTANZ-NETZKNOTEN-LISTE
312
DIENSTGRUPPE 302
ZU HA-SEKUNDÄR
ZUM SLAVE

**FIG. 3**

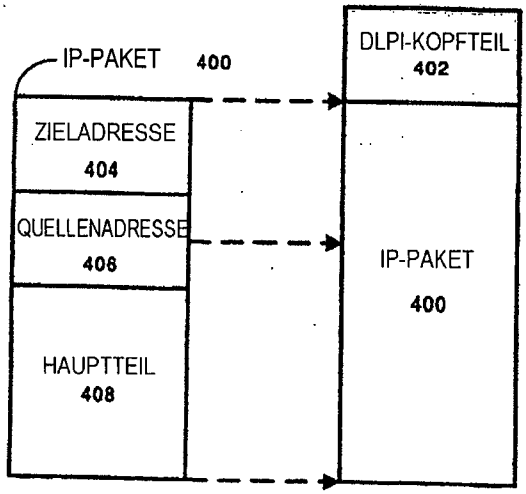
IP-PAKET 400
ZIELADRESSE
404
QUELLENADRESSE
406
HAUPTTEIL
408
DLPI-KOPFTEIL
402
IP-PAKET
400

**FIG. 4**

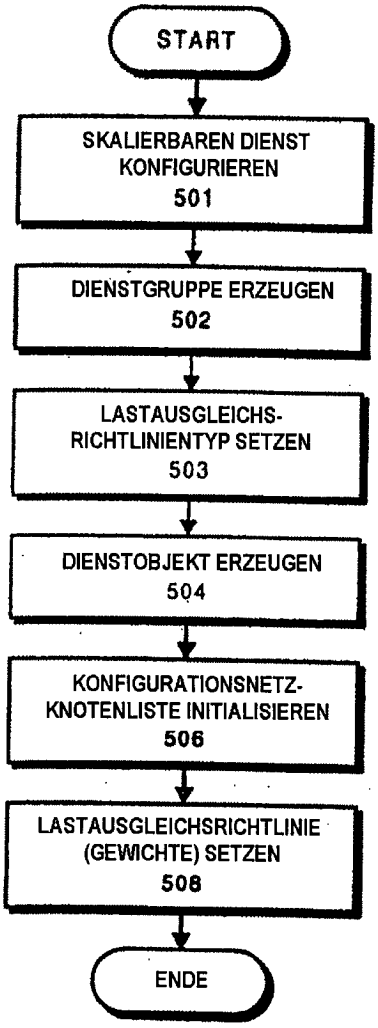
START
SKALIERBAREN DIENST KONFIGURIEREN
501
DIENSTGRUPPE ERZEUGEN
502
LASTAUSGLEICHS-RICHTLINIENTYP SETZEN
503
DIENSTOBJEKT ERZEUGEN
504
KONFIGURATIONSNETZ-KNOTENLISTE INITIALISIEREN
506
LASTAUSGLEICHSRICHTLINIE (GEWICHTE) SETZEN
508
ENDE

FIG. 5A

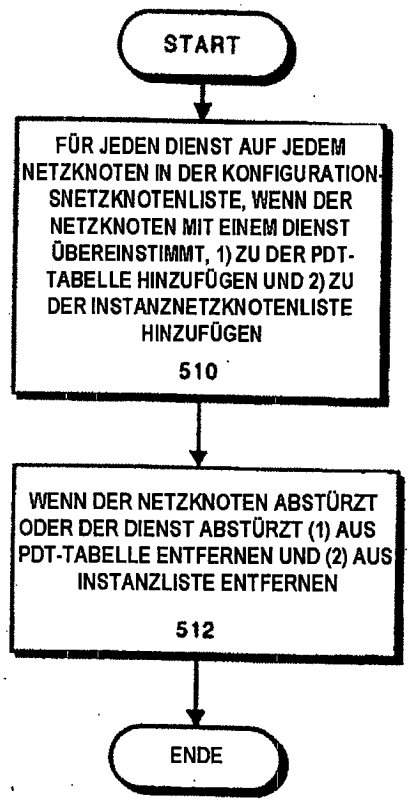
START
FÜR JEDEN DIENST AUF JEDEM NETZKNOTEN IN DER KONFIGURATION-SNETZKNOTENLISTE, WENN DER NETZKNOTEN MIT EINEM DIENST ÜBEREINSTIMMT, 1) ZU DER PDT-TABELLE HINZUFÜGEN UND 2) ZU DER INSTANZNETZKNOTENLISTE HINZUFÜGEN
510
WENN DER NETZKNOTEN ABSTÜRZT ODER DER DIENST ABSTÜRZT (1) AUS PDT-TABELLE ENTFERNEN UND (2) AUS INSTANZLISTE ENTFERNEN
512
ENDE

FIG. 5B

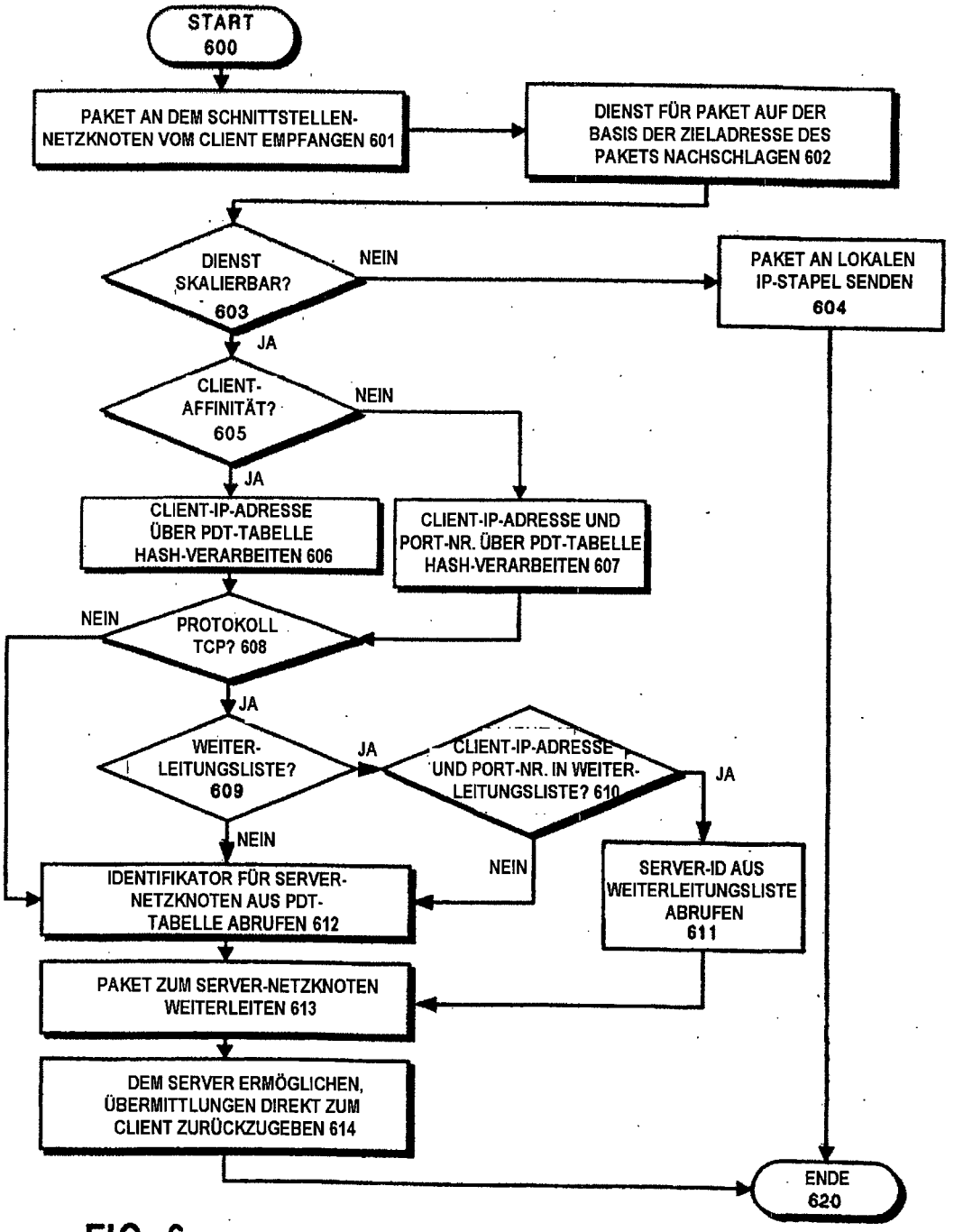
START
600
PAKET AN DEM SCHNITTSTELLEN-NETZKNOTEN VOM CLIENT EMPFANGEN 601
DIENST FÜR PAKET AUF DER BASIS DER ZIELADRESSE DES PAKETS NACHSCHLAGEN 602
DIENST SKALIERBAR? 603
NEIN
PAKET AN LOKALEN IP-STAPEL SENDEN 604
JA
CLIENT-AFFINITÄT? 605
NEIN
JA
CLIENT-IP-ADRESSE ÜBER PDT-TABELLE HASH-VERARBEITEN 606
CLIENT-IP-ADRESSE UND PORT-NR. ÜBER PDT-TABELLE HASH-VERARBEITEN 607
NEIN
PROTOKOLL TCP? 608
JA
WEITER-LEITUNGSLISTE? 609
JA
CLIENT-IP-ADRESSE UND PORT-NR. IN WEITER-LEITUNGSLISTE? 610
JA
NEIN
NEIN
IDENTIFIKATOR FÜR SERVER-NETZKNOTEN AUS PDT-TABELLE ABRUFEN 612
SERVER-ID AUS WEITERLEITUNGSLISTE ABRUFEN 611
PAKET ZUM SERVER-NETZKNOTEN WEITERLEITEN 613
DEM SERVER ERMÖGLICHEN, ÜBERMITTLUNGEN DIREKT ZUM CLIENT ZURÜCKZUGEBEN 614
ENDE
620

FIG. 6

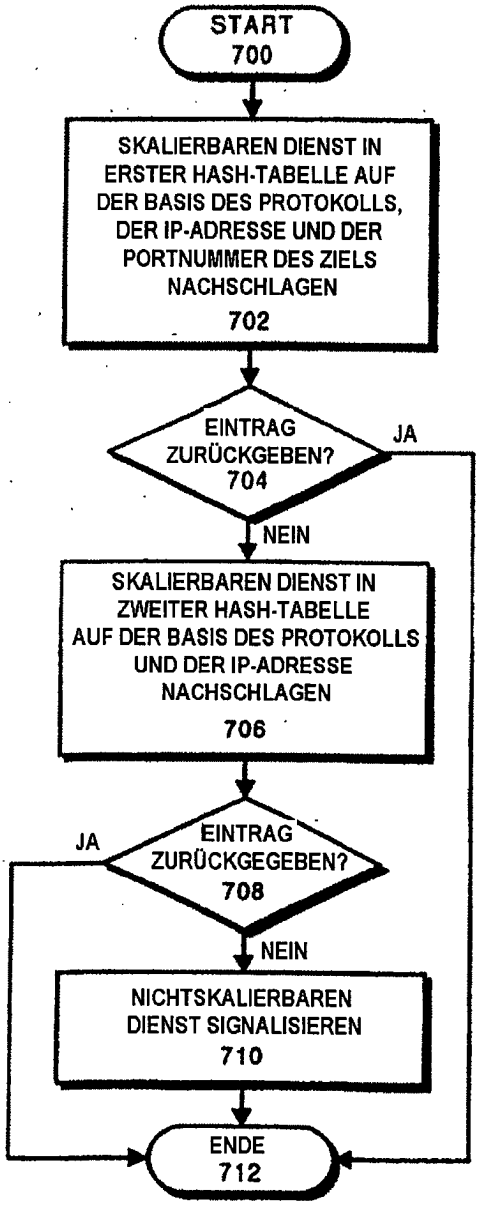
START
700
SKALIERBAREN DIENST IN ERSTER HASH-TABELLE AUF DER BASIS DES PROTOKOLLS, DER IP-ADRESSE UND DER PORTNUMMER DES ZIELS NACHSCHLAGEN
702
EINTRAG ZURÜCKGEBEN?
704
JA
NEIN
SKALIERBAREN DIENST IN ZWEITER HASH-TABELLE AUF DER BASIS DES PROTOKOLLS UND DER IP-ADRESSE NACHSCHLAGEN
706
EINTRAG ZURÜCKGEGEBEN?
708
JA
NEIN
NICHTSKALIERBAREN DIENST SIGNALISIEREN
710
ENDE
712

FIG. 7

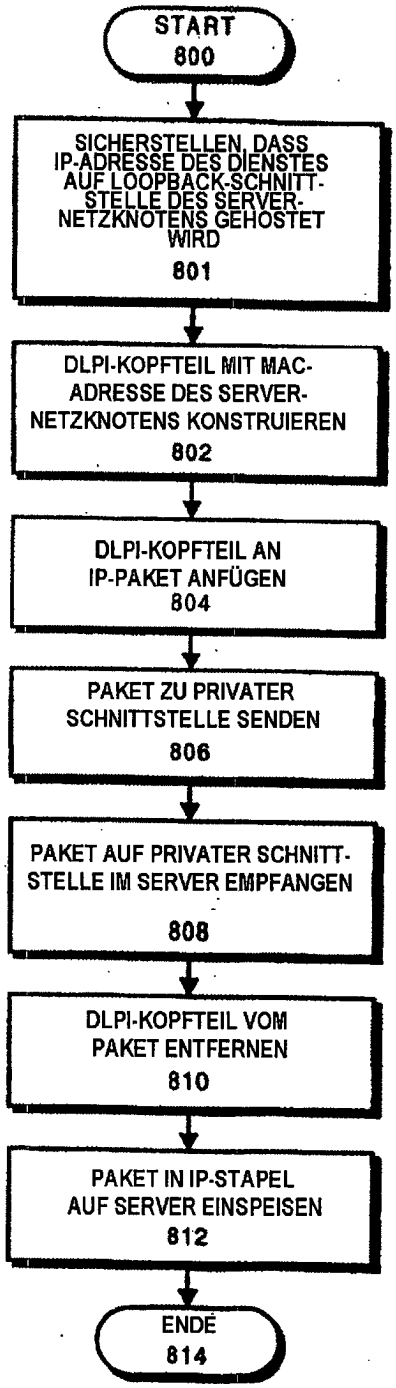
START
800
SICHERSTELLEN, DASS IP-ADRESSE DES DIENSTES AUF LOOPBACK-SCHNITTSTELLE DES SERVER-NETZKNOTENS GEHOSTET WIRD
801
DLPI-KOPFTEIL MIT MAC-ADRESSE DES SERVER-NETZKNOTENS KONSTRUIEREN
802
DLPI-KOPFTEIL AN IP-PAKET ANFÜGEN
804
PAKET ZU PRIVATER SCHNITTSTELLE SENDEN
806
PAKET AUF PRIVATER SCHNITTSTELLE IM SERVER EMPFANGEN
808
DLPI-KOPFTEIL VOM PAKET ENTFERNEN
810
PAKET IN IP-STAPEL AUF SERVER EINSPEISEN
812
ENDE
814

FIG. 8

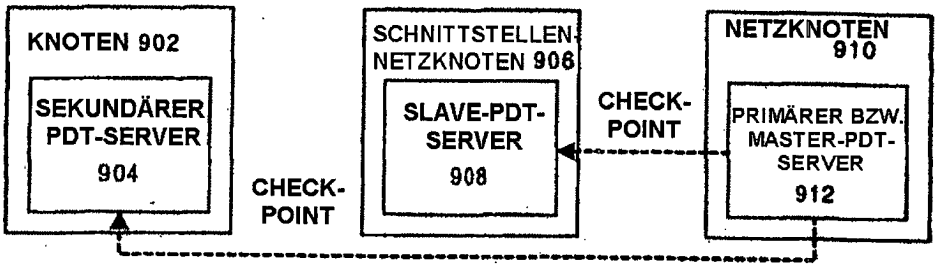
KNOTEN 902
SEKUNDÄRER PDT-SERVER 904
CHECK-POINT
SCHNITTSTELLEN-NETZKNOTEN 906
SLAVE-PDT-SERVER 908
CHECK-POINT
NETZKNOTEN 910
PRIMÄRER BZW. MASTER-PDT-SERVER 912

FIG. 9

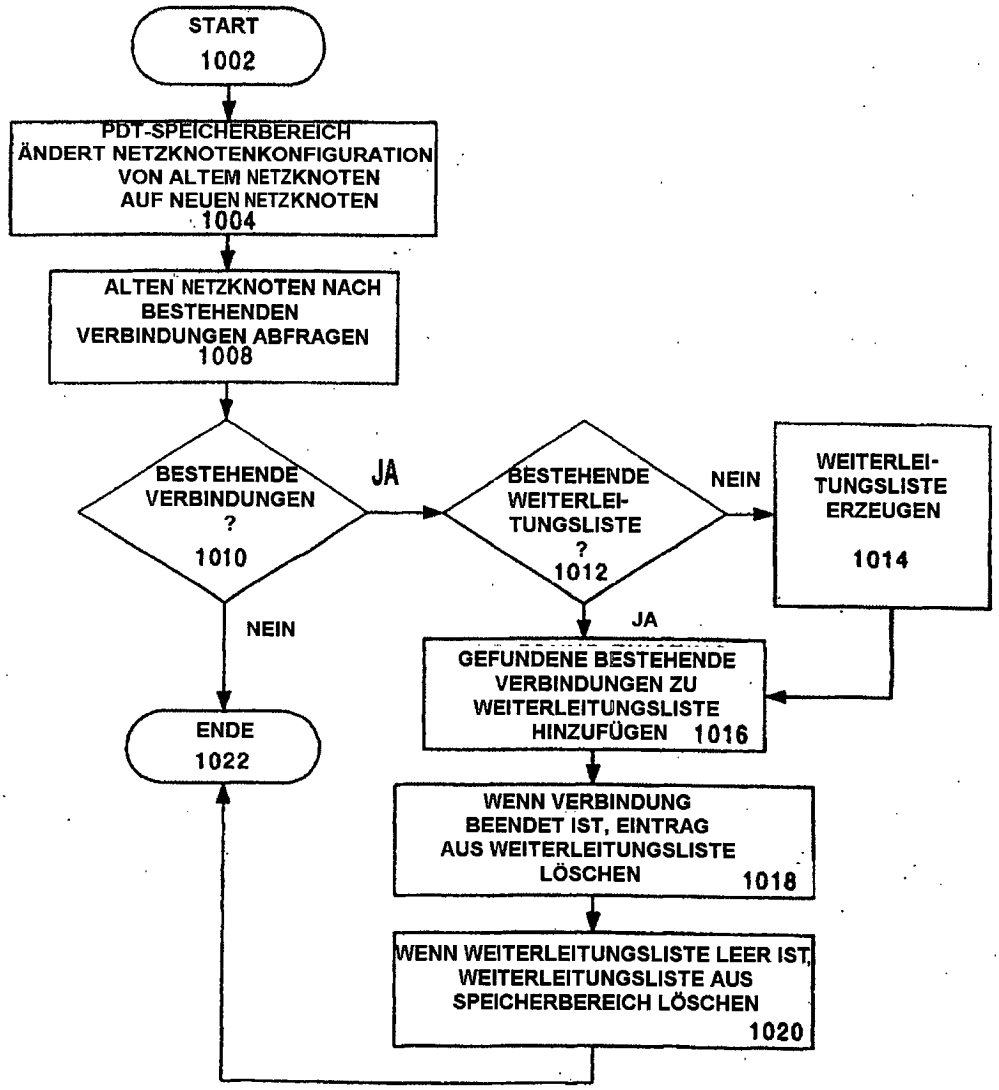
START
1002
PDT-SPEICHERBEREICH ÄNDERT NETZKNOTENKONFIGURATION VON ALTEM NETZKNOTEN AUF NEUEN NETZKNOTEN
1004
ALTEN NETZKNOTEN NACH BESTEHENDEN VERBINDUNGEN ABFRAGEN
1008
BESTEHENDE VERBINDUNGEN ?
1010
JA
BESTEHENDE WEITERLEITUNGSLISTE ?
1012
NEIN
WEITERLEITUNGSLISTE ERZEUGEN
1014
NEIN
JA
GEFUNDENE BESTEHENDE VERBINDUNGEN ZU WEITERLEITUNGSLISTE HINZUFÜGEN
1016
ENDE
1022
WENN VERBINDUNG BEENDET IST, EINTRAG AUS WEITERLEITUNGSLISTE LÖSCHEN
1018
WENN WEITERLEITUNGSLISTE LEER IST, WEITERLEITUNGSLISTE AUS SPEICHERBEREICH LÖSCHEN
1020

FIG. 10