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

(10) **DE 600 22 802 T2** 2006.07.13

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 032 175 B1**
(21) Deutsches Aktenzeichen: **600 22 802.9**
(96) Europäisches Aktenzeichen: **00 300 010.6**
(96) Europäischer Anmeldetag: **05.01.2000**
(97) Erstveröffentlichung durch das EPA: **30.08.2000**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **28.09.2005**
(47) Veröffentlichungstag im Patentblatt: **13.07.2006**

(51) Int Cl.[8]: ***H04L 29/06*** (2006.01)

(30) Unionspriorität:
226864 **07.01.1999** **US**

(84) Benannte Vertragsstaaten:
DE, FR, GB

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

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

(72) Erfinder:
Gasior,Wojciech, Dover, US; Hughes,Aaron, Hayward, US

(54) Bezeichnung: **System und Verfahren zum Übertragen verteilter Datensätze über mehrere Fäden**

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

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

Beschreibung

HINTERGRUND DER ERFINDUNG

[0001] Die vorliegende Erfindung bezieht sich allgemein auf Datenbankabrufsysteme und genauer gesagt auf das Aufteilen großer Datensätze für die Übertragung über mehrsträngige Pfade durch paralleles Laden.

[0002] Datenbanken stellen in vielen Anwendungen ein wichtiges Datenverarbeitungsgebiet dar. Sie werden oft verwendet, um große Mengen an Information zu speichern und dann in effizienter Weise ausgewählte Bereiche der Information aufzurufen. Viele derzeitige Systeme verwenden einen Datenbankansatz von Client/Server, bei welchem Anwendungsprogramme, die auf Clients laufen, auf ein zentrales Datenbankverwaltungssystem (DBMS) zugreifen, das in einem Datenbankserver liegt. Um in effizienter Weise auf die Information in der Datenbank zuzugreifen, bilden die Clients Anfragen, die Informationen mit ausgewählten Eigenschaften anfordern. Diese Anfragen werden an den DBMS-Server übermittelt, der die gewünschte Information aufruft, welche die in der Anfrage spezifizierten Eigenschaften erfüllt. Die Ergebnisse (üblicherweise als „Ergebnissatz" bezeichnet, werden an die Clients zurückgeliefert.

[0003] Die Fähigkeit, Anfragen für das Anfordern von Information in der Datenbank zu bilden, ermöglicht es einem individuellen Benutzer, die Datenbank nach der gewünschten Information zu befragen. In anderen Fällen ist es jedoch wünschenswert, in der Lage zu sein, große Blöcke an Daten in der Datenbank an einen gegebenen Client oder von einem Client zu einem gegebenen Server zu übertragen. Die Übertragung großer Blöcke von Daten innerhalb eines Datenbankverwaltungssystems von einem Client zu einem Server kann typischerweise über mehrere verschiedene Arten von Verbindungssystemen durchgeführt werden. Ein Typ eines solchen Systems ist ein einfaches Verbinden über eine Telefonleitung, wie z. B. ein Einwählen unter Verwendung eines Modems, wobei der Server den Client anwählt und damit fortfährt, den Client nach dem Datenblock zu fragen und dann die Daten über die Modemverbindung auf den Server übertragen lässt. Dieses System sieht einen seriellen Datenübertragungsansatz vor und kann in Anbetracht der begrenzten Übertragungsraten, die derzeit bei Datenverbindungen über Telefon verfügbar sind, ziemlich langsam sein. Ein zweiter Typ an Übertragung kann ausgeführt werden über ein Netzwerksystem mit einem privaten Netzwerk, welches auch als Intranet bekannt ist, wobei der Client und der Server über das Netzwerk miteinander verbunden sind, um eine relativ augenblickliche Kommunikation zwischen den Mitgliedern des Netzwerks zu ermöglichen. Ein Kommunikationsknoten verbindet den Client mit dem Server innerhalb des Netzwerksystems, um einen Strang, bzw. Verbindungspfad (A. d. Ü.: wörtlich: Faden) zu bilden. Das System ist typischerweise viel schneller als die Modemleitung der Telefonverbindung, wegen der hohen Übertragungsraten, die innerhalb von Netzwerksystemen erhältlich sind. Dieser Ansatz einer „einsträngigen" (einfädigen) oder seriellen Übertragung für das Übertragen großer Blöcke von Daten an einen anfordernden Server können bei der derzeitigen Datenübertragung immer noch ziemlich zeitraubend sein wegen der seriellen Natur der Übertragung.

[0004] Ein weiteres System, welches eine Datenbank und Zugriff auf die Datenbank für eine Mehrzahl von Benutzern bereitstellt, ist ein weiträumiges Netzwerk (Wide Area Network – WAN), welches auch innerhalb eines Intranetsystems liegt. Ein WAN enthält typischerweise mehrere Verbindungsknoten, welche mehrere Verbindungen von einer Quelle zu einer anderen ermöglichen. Es werden jedoch selten durch Anwendungen mehrere Verbindungen verwendet, um zwischen einem Client und einem Server innerhalb eines WAN zu kommunizieren. Da die vollständige Kapazität des Netzwerkes durch die Anwendung nicht vollständig ausgenutzt wird, ist die Übertragungsrate immer noch langsam. Es gibt kein bekanntes Verfahren oder System, welches ermöglicht, daß ein großer Datensatz, wie z. B. ein großer Bereich der Datenbank, wenn nicht die gesamte Datenbank, über die Mehrzahl von Knoten in paralleler Weise übertragen wird.

[0005] Daher besteht ein Bedarf daran, in der Lage zu sein, große Datensätze über ein Kommunikationssystem, welches mehrere Knoten hat, über die mehreren Knoten zu übertragen. Was dann benötigt wird, ist ein System oder eine Verfahren, welches ermöglicht, daß Daten über mehrere Knoten in paralleler Weise übertragen werden.

[0006] In „The Parallel I/O Architecture of the High-Performance Storage System (HPSS)" von Watson und Coyne, in Proceedings of the Fourteenth IEEE Symposium on Mass Storage Systems, 11. bis 14. September 1995, Monterey, California, USA, wird die Verwendung paralleler Datenübertragungstechniken für den Clientzugriff auf Speicher, Übertragungen von Peripherie zu Peripherie und Übertragungen entfernt gelegener Dateien beschrieben.

ZUSAMMENFASSUNG DER ERFINDUNG

[0007] Besondere und bevorzugte Aspekte der Erfindung werden in den beigefügten unabhängigen und abhängigen Ansprüchen dargelegt. Merkmale der abhängigen Ansprüche können mit denen der unabhängigen Ansprüche in geeigneter Weise kombiniert werden und auch in anderen Kombinationen, als denjenigen, die in den Ansprüchen ausdrücklich dargelegt sind.

[0008] Gemäß der vorliegenden Erfindung werden ein Verfahren und ein System zum Aufbrechen eines Datensatzes in eine Mehrzahl von Abschnitten bzw. Bruchstücken für die parallele Übertragung über eine Mehrzahl von Kommunikationsknoten bereitgestellt. Das System weist eine Mehrzahl von Kommunikationsknoten auf, die mit den ersten und zweiten Computerspeicherorten verbunden sind. Jeder aus der Mehrzahl von Kommunikationsknoten dient als ein Kommunikationsstrang bzw. -faden. Eine Datenübertragungssteuerung wird mit dem ersten Computerspeicherort und der Mehrzahl von Kommunikationsknoten verbunden. Die Datenübertragungssteuerung wählt eine Anzahl von Kommunikationssträngen aus, damit sie als Datenübertragungsglieder zwischen der ersten und der zweiten Computerspeicherstelle dienen. Ein Datenaufteiler ist ebenfalls vorgesehen, welcher mit dem ersten Computerspeicherort verbunden ist. Der Datenaufteiler reagiert auf eine Datensatzübertragungsanforderung, dadurch, daß er den Datensatz in übertragbare Datensatzabschnitte aufteilt. Der Datensatz besteht aus Datenzeilen, die einen Gesamtbereich haben, so daß eine Abschnittsspalte für die Markierung jedes Datensatzabschnittes bereitgestellt wird, wobei ein Teilbereich von Datenreihen für die Definition jedes Datensatzabschnittes ausgewählt wird und der Teilbereich von Datenreihen einem gegebenen Datensatzabschnitt zugeordnet wird. Die Datenübertragungssteuerung überträgt die Datensatzabschnitte über die Kommunikationsstränge parallel, sobald die Abschnitte bereit sind.

[0009] In einer ersten Ausführungsform wird der Datensatz in eine Anzahl von Datensatzabschnitten aufgeteilt, welche gleich der Anzahl ausgewählter Kommunikationsstränge ist. Dies ermöglicht, daß die Datensatzabschnitte mit den Kommunikationssträngen 1:1 korrespondieren.

[0010] In einer zweiten Ausführungsform wird der Datensatz in einer Anzahl von Datensatzabschnitten aufgeteilt, die größer ist als die Anzahl der ausgewählten Kommunikationsstränge, und sie werden nicht nur parallel über die mehreren Stränge, sondern auch seriell hintereinander verarbeitet, wobei die schnelleren Kommunikationsstränge in der Lage sind, die zusätzlichen Datenabschnitte zu handhaben.

[0011] Ein zusätzliches Merkmal der verteilten Computerumgebung ist die Fähigkeit der Datenübertragungssteuerung bzw. des Datenübertragungscontrollers, Zugriff auf einen übertragenen Datensatzabschnitt zu gewähren, bevor die übrigen Datensatzabschnitte vollständig übertragen und dann zu dem ursprünglichen Datensatz wieder zusammengesetzt sind. Dies ermöglicht es einem Benutzer, auf die Daten innerhalb irgendeines Datensatzabschnittes, der vollständig übertragen worden ist, vor dem eigentlichen Zusammensetzen des ursprünglichen Datensatzes zuzugreifen. Während der Datenaufteilung definiert der Datenaufteiler jeden Abschnitt auf der Basis der Anzahl der Datenzeilen, die innerhalb des Datensatzes verwendet werden. Der Datenaufteiler teilt auch die Datenzeilen zwischen den Abschnitten so gleichmäßig wie möglich auf, ohne irgendwelche der Datenzeilen zwischen getrennten Abschnitten aufzuteilen bzw. zu fragmentieren.

[0012] Das Verfahren wird verwendet innerhalb einer verteilten Computerumgebung, welche einen ersten Computerspeicherort, einen zweiten Computerspeicherort und eine Mehrzahl von Kommunikationsknoten aufweist, die die ersten und zweiten Computerspeicherorte miteinander verbinden. Das Verfahren sieht das Übertragen eines Datensatzes zwischen den ersten und zweiten Computerspeicherorten vor und läuft folgendermaßen ab. Es wird eine Anfrage nach einem Datensatz oder einem Teil eines Datensatzes gemacht, der zwischen den ersten und zweiten Computerspeicherorten übertragen werden soll. Der Datensatz besteht aus Datenzeilen bzw. -reihen, welche einen Gesamtbereich haben bzw. umfassen, so daß eine Aufteilungsspalte für das Markieren jedes Datensatzabschnittes bereitgestellt wird, wobei ein Teilbereich von Datenzeilen, der jeden Datensatzabschnitt definieren soll, ausgewählt wird, und wobei der Teilbereich von Datenzeilen einem gegebenen Datensatzabschnitt zugeordnet wird. Wenn der Datensatz festgelegt ist, teilt der Prozess den Datensatz in übertragbare Datensatzabschnitte auf. Der Prozess bildet eine Mehrzahl von gleichzeitigen Kommunikationssträngen, die aus der Mehrzahl von Kommunikationsknoten ausgewählt werden. Sobald die Abschnitte gebildet worden und die Kommunikationsstränge bereitgestellt sind, überträgt der Prozess die Datensatzabschnitte parallel über die Mehrzahl von Kommunikationssträngen zwischen den ersten und zweiten Computerspeicherorten. Wenn die Kommunikationsstränge gebildet werden, kann die Anzahl der Kommunikationsstränge gleich der Anzahl von Datensatzabschnitten gewählt werden. In ähnlicher Weise kann die Anzahl von Datensatzabschnitten die tatsächliche Anzahl von verfügbaren Kommunikationssträngen übersteigen.

[0013] Wenn die Anzahl von Datensatzabschnitten die Anzahl von Kommunikationssträngen übersteigt, wird über jeden Kommunikationsstrang jeweils ein Abschnitt parallel übertragen. Nur wenn die Datensatzübertragung innerhalb eines Abschnittes abgeschlossen ist und der übertragende Strang verfügbar wird, wird eine weitere Übertragung eines Datensatzabschnittes über den nun verfügbaren Kommunikationsstrang ausgeführt. Die Fähigkeit, sowohl parallele als auch serielle Übertragungen von Datensatzabschnitten durchzuführen, ermöglicht es dem Verfahren, die Übertragungsrate des aufgeteilten Datensatzes auf der Basis der Anzahl von verfügbaren Kommunikationssträngen und der unterschiedlichen Übertragungsgeschwindigkeiten für jeden Kommunikationsstrang zu optimieren.

[0014] Das Verfahren ermöglicht einem Benutzer oder System auch, Zugriff auf vollständige übertragene Datensatzabschnitte zu erlangen, bevor die vollständige Übertragung aller Datensatzabschnitte und ihre Wiederzusammensetzung zu dem ursprünglichen Datensatz erfolgt ist. Der Teilbereich von Datenzeilen für jeden Datensatzabschnitt wird so definiert, daß er keinen anderen Teilbereich von Datenzeilen für irgendeinen anderen Datensatz überlappt. Dies verhindert eine Fragmentierung von Datenzeilen zwischen den Abschnitten, die anderenfalls die Daten beschädigen würde. Der Teilbereich wird in einer solchen Weise ausgewählt, daß er im wesentlichen gleichmäßig verteilte Größen bereitstellt, jedoch ohne exakt identische Größen zwischen den Abschnitten. Weiterhin sieht das Verfahren die Wiederzusammensetzung der Datensatzabschnitte zu dem ursprünglichen Datensatz vor, nachdem die Übertragung abgeschlossen worden ist. Die vorliegende Erfindung enthält weiterhin ein Computerprogrammprodukt für die Verwendung in einem verteilten Computersystem, wie es zuvor beschrieben wurde. Das Computerprogrammprodukt enthält einen Computerprogrammcode, der in einer Art und Weise ähnlich dem Verfahren der vorliegenden Erfindung arbeitet. Das Computerprogrammprodukt ermöglicht, daß eine Softwareimplementierung der Erfindung auf einem verteilten Computersystem, wie z. B. einem LAN oder WAN oder dem Internet ausgeführt wird.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0015] Nachstehend werden beispielhafte Ausführungsformen der Erfindung lediglich in Form von Beispielen beschrieben, und zwar unter Bezug auf die beigefügten Figuren, von denen:

[0016] Fig. 1 ein schematisches Blockdiagramm eines Client-Server-Systems ist, auf welchem die vorliegende Erfindung ausgeführt werden kann,

[0017] Fig. 2 ein schematisches Blockdiagramm ist, welches die Wechselwirkung zwischen Clientobjekt, Geschäftsobjekten und Anfrageobjekten in einem dreistufigen System veranschaulicht,

[0018] Fig. 3 ein schematisches Diagramm ist, welches eine Zuordnungstabelle für das Zuordnen von sortierten Datenzeilen innerhalb eines Datensatzes zu ausgewählten Abschnitten für die nachfolgende Übertragung veranschaulicht,

[0019] Fig. 4 ein Flussdiagramm der Schritte zeigt, die beim Bereitstellen der Anzahl von Abschnitten innerhalb eines Datensatzes verwendet werden, der übertragen werden soll, ebenso wie die Anzahl von Kommunikationssträngen, die verwendet werden, um die Übertragung von Datensatzabschnitten gemäß der vorliegenden Erfindung durchzuführen.

GENAUE BESCHREIBUNG DER SPEZIELLEN AUSFÜHRUNGSFORM

[0020] Fig. 1 ist ein Blockdiagramm eines Systems **100**, auf welchem ein Datensatzverteilungswerkzeug, welches gemäß der vorliegenden Erfindung aufgebaut ist, ablaufen kann. Wie dargestellt, ist das System **100** eine verteilte Rechnerumgebung, welche eine Mehrzahl einzelner Computerknoten **102**, **104**, **106** und **108** aufweist. Die Knoten sind funktionell organisiert als Clients **102**, **104** und **106** und zumindest ein Server **108**, welche über das Netzwerk **105** miteinander verbunden sind. Jeder Knoten, sei es ein Client **102**, **104**, **106** oder ein Server **108**, ist ein konventionell programmierter, digitaler Computer, der jeweils eine zentrale Verarbeitungseinheit (CPU) **118**, **122** und einen Hauptspeicher **112**, **124** enthält, welcher mit der CPU **118**, **122** verbunden ist. Zusätzlich kann jeder Client **102**–**106** vom Server **108** Benutzerschnittstelleneinrichtungen, wie z. B. einen Monitor **114**, eine Tastatur **116**, eine Maus **117** und eine Speichereinrichtung **112** (in Fig. 1 beispielhaft nur für einen Client **102** dargestellt) enthalten. Der Server **108** enthält eine Datenbank **110**, wie z. B. eine relationale Datenbank, ein Dateisystem oder ein anderes organisiertes Datenspeichersystem. Der zentrale Systembus **105** enthält eine Mehrzahl von Kommunikationsknoten **126**, **128** und **130**, welche Kommunikationswege zwischen einem Client **102**–**106** und dem Server **108** oder zwischen einem ersten und einem zweiten Client **102**–**106** bereitstellen. Schließlich kann das System **100** Einrichtungen enthalten, um computerlesbare Spei-

chermedien (nicht dargestellt) anzunehmen, und kann mit dem Netzwerk **105** verbunden sein, kann mit dem Netzwerk **105** verbunden sein, um Daten- und Steuersignale auszutauschen, welche als ein Computerdatensignal in einer Trägerwelle übermittelt werden.

[0021] In der beschriebenen Ausführungsform ist jeder Knoten **102**–**108** ein mit einem Netzwerk verbindbarer Computer, wie z. B. eine Sun SparcStation™5 workstation, die unter dem Solaris™-Betriebssystem läuft, eine Version des UNIX®-Betriebssystems, oder ein IBM-kompatibler Computer, der unter dem Windows NT™-Betriebssystem läuft. Die Verwendung der Systeme und Prozesse, die hier beschrieben und vorgeschlagen werden, sind jedoch nicht auf eine bestimmte Computerkonfiguration beschränkt. SparcStation™ und Solaris™ sind Marken von Sun Microsystems, Inc., Mountain View, Kalifornien. UNIX® ist eine registrierte Marke von Santa Cruz Operation, Santa Cruz, Kalifornien, Windows NT™ ist eine Marke von der Microsoft Corporation in Redmond, Washington.

[0022] Jeder Kommunikationsknoten **126**–**130** ist ein Netzwerkübertrager, welcher Datenpakete, die von einem Knoten **102**–**108** an einen anderen ähnlichen Knoten gesendet werden, empfängt. Das Netzwerksystem leitet typischerweise Datenpakete auf der Basis des zeitlich schnellsten Pfades zwischen dem anfordernden Knoten und dem empfangenden Knoten **102**–**108**. Da eine Mehrzahl von Kommunikationsknoten **126**–**130** bereitgestellt ist, ist es möglich, mehrere Knoten zu verwenden, wobei jeder Knoten einen einzelnen Kommunikationsstrang repräsentiert, um ein mehrsträngiges Übertragungssystem gemäß der vorliegenden Erfindung zu bilden. Das System **100** kann ein großräumiges Netzwerk (WAN) sein, in welchem der Anwendungsclient und der Datenbankserver durch viele Knoten **126**–**130** oder hops getrennt sein können, die eine hohe Netzwerklatenz haben.

[0023] Weiterhin kann jeder Knoten **102**–**108** als eine Datenübertragungssteuerung **132** und als ein Datenaufteiler **134** arbeiten. Die Datenübertragungssteuerung **132** stellt die Gesamtzahl von Kommunikationssträngen bereit, die verwendet werden können, um eine parallele Übertragung eines aufgeteilten Datensatzes zwischen ausgewählten Knoten bereitzustellen, ebenso wie die eigentliche Übertragung der Abschnitte über die Stränge. Der Knoten, der entweder der übertragende oder der empfangende Knoten ist, kann entweder als Datenübertragungssteuerer **132**, als Datenaufteiler **134** oder als beides dienen. Der Datenaufteiler **134** zerschneidet den zu übertragenden Datensatz und stellt auch die Wiederzusammensetzung des Datensatzes bereit, nachdem alle aufgeteilten Abschnitte des Datensatzes erfolgreich an den gewünschten Knoten übertragen wurden.

[0024] Verteilte Rechnerumgebungen können logisch als ein Satz von zusammenwirkenden Softwarekomponenten betrachtet werden, die als „Objekte" bezeichnet werden, und die auf einem oder mehreren Computern ausgeführt werden, welche durch ein Netzwerk verbunden sind. Die individuellen Anwendungen, die auf jedem Computer laufen, können einen Satz von standardmäßigen oder anwendungsspezifischen Objekten gemeinsam verwenden und das Ausführen einer Anwendung kann dazu führen, daß sie mit anderen Softwareobjekten, die lokal und über das Netzwerk verfügbar sind, in Wechselwirkung tritt und diese verwendet. Jedes derartige Objekt kann seinerseits andere Softwareobjekte verwenden. Demnach kann eine Anwendung in einer verteilten Rechnerumgebung aus der Kombination von anwendungsspezifischen Objekten und den kombinierten lokalen und entfernten Objekten bestehen, die sie verwendet. Wenn ein Objekt einen Vorgang bzw. eine Operation auf einem anderen Objekt aufruft, wird das aufrufende Objekt als der „Client" bezeichnet, während das angerufene Objekt als der „Server" bezeichnet wird. Fig. 2 ist ein schematisches Blockdiagramm einer Client/Server-Umgebung **200** einer Datenbank, welche Datenanforderungsobjekte **208** und **214** verwendet, wie sie möglicherweise in dem System **100** nach Fig. 1 implementiert sein könnten. Die Umgebung **200** ist funktionell in drei Stufen aufgebaut. In der obersten Stufe **203** führt ein Client **102** (dargestellt in Fig. 1) eine Anwendung **202** aus. Die Anwendung **202** steht in Wechselwirkung mit der Geschäftslogik in der mittleren Stufe **205**, wie es durch den Pfeil **204** schematisch angedeutet wird. Die Geschäftslogik ist in Form eines oder mehrere Geschäftsobjekte **206** implementiert (nur ein Geschäftsobjekt **206** ist dargestellt) und sie bildet ihrerseits eine Schnittstelle mit Datenanforderungsobjekten **208** und **214**. Sowohl das Geschäftsobjekt **206** als auch die Datenanforderungsobjekte **208** und **214** können entweder auf einem Client **102** oder einem Server **108** (dargestellt in Fig. 1) ausgeführt werden. Andere Kombinationen und Konfigurationen von Objekten, einschließlich eines Geschäftsobjektes **206** oder von Datenanforderungsobjekten **208** und **214** sind vorstellbar.

[0025] In der untersten Stufe **207** enthalten ein Server **108** und eine zugehörige Datenbank **110** (dargestellt in Fig. 1) Datenbankverwaltungssysteme (DBMS-Systeme) **218** und **222** (welche auf dem Server **108** ausgeführt werden können), und zwar zusammen mit zugehöigen Datenbanken **224** bzw. **226**. Die Datenanforderungsobjekte **208**, **214** ermöglichen es der in dem Geschäftsobjekt **206** implementierten Geschäftslogik, auf die DBMS-Systeme **218**, **222**, zuzugreifen.

[0026] Auch wenn die derzeitigen Systeme in der Lage sind, große Datensätze unbeschädigt über die in dem Client/Server-System bereitgestellten Verbindungen zu übertragen, tun sie dies nur unter Verwendung eines einzelnen Stranges, um den Datensatz in serieller Weise zu übertragen. Was nunmehr präsentiert wird, ist ein Verfahren und ein System, welches ein sehr umfangreiches und schnelles Laden von Datensätzen aus Servern ermöglicht, wie z. B. (ohne jedoch speziell darauf beschränkt zu sein) einem SQL-Server, der eine Mehrzahl von Zeilen aus einer Datenbank heranholt. Die vorliegende Erfindung kann auch auf Anwendungen auf Webbasis anwendbar sein, um einen großen Datensatz über mehrere Knoten heranzuholen.

[0027] In einer Ausführungsform verwendet das System **100** mehrere Stränge, um die Daten aus irgendeinem ferngelegenen DBMS-Ort über eine standardmäßige ANSI SQL-Schnittstelle zu laden. Das System **100** erzielt eine verminderte Gesamtladezeit für das Übertragen dieser großen Datensätze, in dem es die Daten in der Datenbank aufteilt, mehrere Verbindungen mit dem Datenbankserver erzeugt, und jeden Abschnitt parallel über eine Mehrzahl gleichzeitiger Verbindungen lädt, was von dem erhöhten Durchsatz, der durch eine parallele Übertragung geboten wird, profitiert.

[0028] Jede gleichzeitige Verbindung wird durch die Mehrzahl von Knoten **126**–**130** bereitgestellt, welche in dem WAN-System **100** vorgesehen sind. In einem Test erzeugten 10 gleichzeitige Übertragungsverbindungen eine Übertragungsgeschwindigkeit, die zumindest sechs- bis siebenmal so groß wie die Geschwindigkeit einer einzelnen Verbindung war.

[0029] Der ideale Vervielfachungsfaktor der Übertragungsgeschwindigkeit würde n-mal die Geschwindigkeit einer Einzelverbindung betragen, wobei n die tatsächliche Anzahl gleichzeitig bereitgestellter Verbindungen ist; es gibt jedoch eine zusätzliche inhärente Verzögerung beim Öffnen von mehreren Verbindungen. Die Verzögerung der Übertragungsgeschwindigkeit in jedem Strang, der in einer mehrsträngigen Verbindung verwendet wird, resultiert aus einer zusätzlichen Sendeüberlast, die für den Server benötigt wird, um jede Verbindung bereitzustellen, den Strang aufrecht zu erhalten und die übertragenen Daten zu empfangen. Die Übertragungsgeschwindigkeit in einer mehrsträngigen Übertragung ist dennoch schneller als die eines einzelnen Stranges, auch wenn jedem einzelnen Strang die Verzögerung hinzuaddiert wird.

[0030] Weiterhin kann das System eine größere Anzahl von Abschnitten erzeugen, als es Übertragungsstränge gibt. Ein erster Satz von Abschnitten, welcher gleich der Anzahl von bereitgestellten Übertragungssträngen ist, wird parallel über die Übertragungsstränge übertragen. Als nächstes werden, da einige Stränge eine Übertragungsgeschwindigkeit haben, die schneller ist als die anderer Stränge, diejenigen Stränge, die die Übertragung eines ersten Abschnittes beendet haben, dann verwendet, um die verbleibenden Datenabschnitte je nach der Verfügbarkeit der Stränge zu übertragen. Die Tatsache, daß man in der Lage ist, sowohl eine parallele Übertragung des Datensatzes als auch eine serielle Übertragung einer größeren Anzahl von Abschnitten bereitzustellen, ermöglicht eine schnellere Übertragungsgeschwindigkeit in einem System, welches eine begrenzte Anzahl verfügbarer Kommunikationsknoten hat. Zusätzlich optimiert das Mischen von sowohl paralleler als auch serieller Übertragung der Datenabschnitte über die mehreren Kommunikationsstränge, die Übertragungsstränge, welche in dem System am schnellsten arbeiten, um die Übertragungsgeschwindigkeit gegenüber derjenigen eines einzelnen Knotens zu verbessern. Das Verfahren und das System ist in der Lage, die Daten in der Datenbank automatisch und unabhängig von der Anwendungslogik aufzuteilen. Dies ermöglicht es dem System, den Datensatz über die mehreren Stränge parallel zu laden und zu übertragen, und die Ergebnisse zu einem einzelnen Datensatz zusammenzusetzen, was den Eindruck einer schnellen Einzelverbindung hervorruft. Zusätzlich ist ein Wiederzusammensetzen des ursprünglichen Datensatzes vor dem Zugreifen auf irgendwelche Daten in einem der vollständig übertragenen Datenabschnitte bei einer mehrsträngigen Anwendung, die dafür ausgelegt ist, die Daten in größeren Stücken zu übertragen, wohlmöglich nicht notwendig. Insbesondere haben solche Anwendungen zusätzliche Vorteile, indem sie parallele mehrsträngige Übertragungen verwenden, da dieses ermöglicht, daß auf die Daten zugegriffen wird, sobald die Übertragung eines Abschnittes abgeschlossen ist. Wenn also ein gegebener Abschnitt vollständig übertragen worden ist, kann das System auf die Daten in dem Abschnitt zugreifen, ohne darauf warten zu müssen, daß die übrigen Abschnitte die Übertragung beendet haben und dann alle gesendeten Abschnitte als der ursprüngliche Datensatz wieder zusammengesetzt ist.

[0031] Das System und das Verfahren stellen eine Erweiterung **312** für die Anwendungsdatenbank **300** bereit, welche in Fig. 3 dargestellt ist. Die Erweiterung **312** ist unabhängig von der Anwendungslogik und wird für die automatische Aufteilung von Daten in Tabellen verwendet. Die Datenbank **300** enthält gemäß der vorliegenden Erfindung einen Datensatz **310**, der die gesamte Datenbank oder einen großen Ausschnitt der Datenbank **300** enthalten kann, der in der Datenbank **110** oder einem Client **102**–**106** nach Fig. 1 gespeichert sein kann. Jeder Datensatz **310** besteht aus einer Mehrzahl von Datenzeilen **316**. Die Erweiterung **312** umfasst das

Hinzufügen einer neuen Spalte **318**, welche eine „Aufteilungsspalte" genannt wird. Die Aufteilungsspalte **318** wird verwendet, um die speziellen Datenzeilen **318** in einem definierten Datensatz **310** mit ganzzahligen Werten zuzuordnen. Der ganzzahlige Wert wird in der Aufteilungsspalte **318** gespeichert, um die speziellen Datenzeilen **316** für den definierten Abschnitt **314** zu verfolgen. Jede Aufteilungsspalte **318** hat einen Wertebereich <0, PARTITION_MAX_VAL-1>, wobei PARTITION_MAX_VAL eine Zahl ist, die während der Anwendungsinstallation zugeordnet wird. PARTITION_MAX_VAL repräsentiert die maximale Anzahl von Abschnitten, die beim Auseinanderschneiden der Datenbank für eine Übertragung über das WAN-System **100** erlaubt wären. Diese Anzahl wird festgelegt auf Basis der Anzahl der Knoten, welche als unabhängige Kommunikationsstränge für die parallele Übertragung der Abschnitte zu dem Ziel der aufgeteilten Datenbank verwendet werden können. Man kann sich eine alternative Ausführungsform vorstellen, welche die Abschnittsnummern in einer getrennten Tabelle anordnet. Wenn der Datensatz **310** das Ergebnis einer SQL-Verbindungsoperation ist, welche mehrere Datentabellen umfasst, so ist alles, was zur Aufteilung des Satzes erforderlich ist, die Anwesenheit der Aufteilungsspalte **312** in irgendeiner der teilnehmenden Tabellen, die angeschlossen sind.

[0032] Das Verfahren zum Bereitstellen der Abschnitte **314** der Fig. 3 ist in dem Blockdiagramm nach Fig. 4 dargestellt. Das Verfahren beginnt mit Schritt **400** und geht weiter zu Schritt **410**, wo die Anforderung nach einer Übertragung eines Datensatzes **310** empfangen wird und dann Vorbereitungen getroffen werden, um die Anforderung der Datensatzübertragung zu erfüllen. In Schritt **412** bestimmt das System die Anzahl von Aufteilungen, die benötigt wird, um den Datensatz **310** über das WAN-System **100** zu übertragen. Die Entscheidung über die Auswahl der Anzahl von Abschnitten hängt ab von der Größe des zu übertragenden Datensatzes, des aktuell durch das WAN-System **100** getragenen Verkehrs, der für die Übertragung des Satzes in der maximal zulässigen Zeit und der Eile, mit welcher die Daten benötigt werden. Als nächstes sieht in Block **414** eine Erweiterung **312** vor, welche die Aufteilungsspalte **318** bereitstellt.

[0033] Dann erzeugt in Block **416** ein Zufallszahlengenerator gleichmäßig verteilte Werte in dem Bereich <0 bis PARTITION_MAX_VAL-1>. Der Zufallszahlengenerator kann in den Speicherprozeduren der Datenbank oder in dem Java-Code der Anmeldung implementiert sein. Die Zufallszahlen sind in dem Modul verfügbar, welches Zeilen in einer Datenbanktabelle mit Aufteilungsspalten erzeugt. In Block **408** werden die Zufallswerte aus dem gegebenen Bereich der Aufteilungsspalte **318** der neu erzeugten Reihen zugeordnet. Die gleichmäßig verteilten Werte müssen nicht exakt gleich sein, da die Datenzeilen **316** nicht immer die gleiche Länge haben und da es gewünscht ist, daß keine fragmentierten Datenzeilen **316** während des Aufteilens auftreten. Auch wenn die Aufteilungsnummern hier während des Ladevorganges zugeordnet werden, können sie alternativ stattdessen auch während des Schreibvorganges zugeordnet werden.

[0034] Unter Verwendung der Zufalls()methode in der Math-Klasse von Java wird der ganzzahlige Bereich folgendermaßen bestimmt:

```
Int. rn = (int) (java.lang.Math.random ()* (double) PARTITION_MAX_VAL)
```

rn liegt in dem geforderten Bereich <0, PARTITION_MAX_VAL-1>

[0035] Das System ordnet die Zufallswerte der Aufteilungsspalte zu, sobald eine Zeilen in der Zieltabelle erzeugt wird. Der Java- oder SQL-Code der Anwendung, welcher verwendet wird, um Reihen in die Aufteilungstabellen einzufügen, wird modifiziert, um die Zuordnung von Zufallswerten zum Aufteilen jeder Spalte in neue Reihen zu umfassen. Die Zufallszahlen sind insofern zweckmäßig, als daß sie mit hoher Wahrscheinlichkeit sicherstellen, daß die Abschnitte in gleichgroß bemessene Einheiten aufgeteilt werden, wenn sich die Population der Zeilen verändert. Die Zeilenpopulation verändert sich während der Hinzufügung oder Löschung oder während anderer Modifikationen der verschiedenen Zeilen während des Anwendungszugriffes und der Manipulation während des beabsichtigten Gebrauchs des aufgeteilten Datensatzes **310**.

[0036] In Schritt **420** teilt das System Zeilen in den Zieltabellen auf, indem es die Abschnittsspaltenwerte berechnet, die zu denselben Abschnitten gehören, und zwar unter Verwendung der PARTITION_MAX_VAL-Methode und der Anzahl der gewünschten Abschnitte. Dies kann in einem beliebigen SQL-Server folgendermaßen implementiert werden:

<0, PARTITION_MAX_VAL-1> - der Bereich von Zufallswerten
- die Anzahl gewünschter Abschnitte

Seien P, E, K ganze Zahlen, die definiert sind als:

P = PARTITION_MAX_VAL/N ; (man beachte ganzzahlige Aufteilung)

E = P + 1 ;

K = PARTITION_MAX_VAL-P * N ;

Sei <rl, rM> der Aufteilungsbereich, dann gilt

für Abschnitte:

I = 0, 1, ..., K-1

ist der Bereich von Werten

<I * E, I * E + P>

rl = I * E

rM = I * E + P

und für die verbleibenden Abschnitte

I + K, K + 1, ..., N – 1

ist der Bereich von Werten

<I * P + K, I * P + K + P – 1>

rl = I * P + K

rM = I * P + K + P –1

[0037] Bei diesem Vorgang sind, falls K > 0, die ersten K Abschnitte größer und jeder enthält einen zusätzlichen Zufallszahlenwert in dem Bereich. Dies hat nur wenig Einfluß auf die Zeilenzahl in den Abschnitten, wenn PARTITION_MAX_VAL ausreichend groß ist. Ein Beispiel wird folgendermaßen gegeben:

Für PARTITION_MAX_VAL = 1000000 und N = 7

<0,	999999>	- der gesamte Zufallszahlensatz
<0	142857>	- Abschnitt 0
<142858,	285714>	- Abschnitt 1
<285715,	428571>	- Abschnitt 2
<428572,	571428>	- Abschnitt 3
<571429,	714285>	- Abschnitt 4
<714286,	857142>	- Abschnitt 5
<857143,	999999>	- Abschnitt 6

[0038] Jeder Knoten **102**–**108** kann als eine Mehrstrangsteuerung dienen, die verwendet wird, um mehrere gleichzeitige Verbindungen mit der Datenbank zu verwalten, und damit als die Datenübertragungssteuerung **132** dient. Jeder Knoten **102**–**108** ist in der Lage, dynamisch Abschnitte auf der Basis der Anzahl von verfügbaren Verbindungen zu berechnen und, in Block **422**, den parallelen Ladevorgang von Datenabschnitten **314** für die Übertragung von dem Client zu dem Server oder umgekehrt auszuführen. Demnach dient jeder Knoten **102**–**108** als Datenaufteiler. Da die Abschnitte **314** gleichzeitig geladen werden, können sie, falls notwenig, wieder zu einem Datensatz **10** zusammengesetzt werden und für die weitere Verarbeitung in der Anwendung übergeben werden, sobald sie von dem anfordernden Server empfangen wurden. Wenn ein gegebener Abschnitt geladen worden ist, kann er für den Benutzer verfügbar gemacht werden, wie es in Schritt **424** dargestellt ist. In Schritt **426** sind die Abschnitte wieder zu dem ursprünglichen Datensatz zusammengesetzt worden, nachdem auf den gegebenen Abschnitt zugegriffen worden ist, falls dies angefordert wurde und nachdem alle verbleibenden Abschnitte geladen worden sind. Jeder Knoten **102**–**108** ist in der Lage, jeglichen gewünschten

Wert einer Abschnittsgröße zu handhaben. Für kleine Abschnittswerte verwendet das System eine erste Implementierung. Für größere Abschnittswerte verwendet das System eine alternative Implementierung. Die erste Implementierung für sehr kleine Abschnitte, mit einem Wert von 32 und darunter, wobei jedoch jede zweiziffrige Zahl in den sehr kleinen Abschnittswertebereich fällt, wird nachstehend wiedergegeben:

Für einen sehr kleinen PARTITION_MAX_VAL-Wert

wähle.....

von.....

wobei.....

(Abschnitt = r1 oder Abschnitt = r2…oder Abschnitt = rM)

… - zeigt den anwendungsspezifischen Code an.

Wobei <r1, rM> der Abschnittsbereich und r1, r2,...rM aufeinanderfolgende ganze

Zahlen in dem Bereich sind.

[0039] Das alternative Verfahren zum Durchführen eines Aufteilungsvorganges wird folgendermaßen bereitsgestellt:

Für irgendeinen PARTITION_MAX_VAL-Wert

wähle.....

von.....

wo.....

(Abschnitt >= r1 und Abschnitt <+ rM)

... – zeigt den anwendungsspezifischen Code an.

Wobei <rl, rM> der Abschnittsbereich ist.

rl und rM werden folgendermaßen berechnet:

rl = I * P + K

rM = I * P + K + P –1

[0040] Die Implementierung mit dem sehr keinen Abschnitts- bzw. Aufteilungswert verwendet eine Indexierung an der abgetrennten Spalte und kann effizienter sein als die alternative Implementierung.

[0041] Es wurde ein Übertragungstest durchgeführt, bei welchem die Knoten **102**–**108** verwendet wurden, um einen parallelen Ladevorgang durchzuführen. Der Ladevorgang erreichte eine Übertragungsrate, die zwischen 60 und 70% der maximalen theoretischen Geschwindigkeit der verwendeten Verbindungen betrug. Wenn beispielsweise zehn gleichzeitige Verbindungen existieren, ist die Übertragungsrate etwa siebenmal schneller als es über eine einzige Verbindung möglich wäre. Die Ergebnisse des Übertragungstests, eine Aufteilung, welche drei gleichzeitige Verbindungen verwendete, werden nachstehend wiedergegeben:

tanda: wgasior[119]%java Pload3

Paralleler Ladetest: 3 Abschnitte(s)

_angefordert 3 Abschnitte

Starten der Zeitgeber.....
Erzeugen von Strängen.....
Verstrichene Zeit: 2ms

_benötigte 2ms zum Erzeugen der Aufteilungsstränge

Starten der Stränge.....
Verbinden der Stränge.....

_der Hauptstrang wartet darauf, daß die Aufteilungsstränge das Laden der Daten von einer Datenbank und das Ausliefern der Ergebnisse beenden

Abschnitt 0 – begonnen: Start = 0, Ende = 333333
Abschnitt 1 – begonnen: Start = 333334, Ende = 666666
Abschnitt 2 – begonnen: Start = 666667, Ende = 999999

_jeder Abschnitt wählt alle Reihen in der Datenbank aus, die Werte der Aufteilungsspalte in dem Bereich zwischen <Start, Ende> haben

Abschnitt 1 – beendet: Verzögerung = 3313ms, Abfrage = 278ms, Laden = 89861ms, Zeilen = 1273
Abschnitt 2 – beendet: Verzögerung = 4096ms, Abfrage = 280ms, Laden = 93512ms, Zeilen = 1324
Abschnitt 0 – beendet: Verzögerung = 2376ms, Abfrage = 434ms, Laden = 98853ms, Zeilen = 1403

_delay bzw. Verzögerung ist hauptsächlich die Zeit für das Beschaffen der jdbc-Verbindung (vor der eigentlichen Arbeit)
Abfrage ist die Zeit der Ausführung der ausgewählten Anweisungen (jdbc ausführen ()).
Laden ist die Zeit des Ladens aller Zeilen in einem gegebenen Abschnitt
Zeilen ist die Anzahl der Zeilen in einem gegebenen Abschnitt

Gesamte Zeilenzahl: 4000

_total (gesamt) Zeilen in allen Abschnitten

Insgesamt verstrichene Zeit: 101665ms

_die Gesamtzeit für den Ablauf des vollständigen Programms

[0042] Wie in dem obrigen Beispiel dargestellt, ergibt sich für die Übertragung von 4000 Zeilen innerhalb ei-

nes Datensatzes, der in 3 Abschnitte aufgeteilt worden ist, eine insgesamt verstrichene Zeit von 101665ms. Diese verstrichene Zeit ist schneller als die Übertragung über einen einzelnen Strang und wird nachstehend dargestellt. Die Übertragung über einen einzelnen Strang, die unten wiedergegeben ist, hatte eine insgesamt verstrichene Übertragungszeit von 282348ms. Nach dem Beispiel eines einzelnen Strangs, der verwendet wurde, um den gewünschten Datensatz zu übertragen, werden Testbeispiele von Abschnitten wiedergegeben, die von zwei Abschnitten bis zehn Abschnitten reichen, wobei ihre gesamten Übertragungszeiten angegeben sind. Man erkennt, daß die Übertragung mit dem einzelnen Strang am langsamsten ist, während die Übertragung mit zehn Aufteilungssträngen die kürzeste insgesamt verstrichene Zeit hat.

Ein Abschnitt:

```
tanda:wgasior[117]%java Pload 1

Paralleler Ladetest: 1 Abschnitt(s)
Starten der Zeitgeber.....
Erzeugen der Stränge.....
Verstrichenen Zeit: 1ms
Starten der Stränge.....
Verbinden der Stränge.....
Abschnitt 0 – begonnen: Start = 0, Ende = 999999
Abschnitt 0 – beendet: Verzögerung = 1253ms, Abfrage = 430ms, Laden = 280661ms, Zeilen = 4000
Gesamtzahl der Zeilen: 4000
Insgesamt verstrichene Zeit: 282348ms
```

2 Abschnitte:

```
tanda.wgasior[118]%java Pload 2
Paralleler Ladetest: 2 Abschnitte(s)
Starten der Zeitgeber.....
Erzeugen der Stränge.....
Verstrichenen Zeit: 1ms
Starten der Stränge.....
Verbinden der Stränge.....
```

Abschnitt 0 – begonnen: Start = 0, Ende = 499999

Abschnitt 1 – begonnen: Start = 500000, Ende: = 999999

Abschnitt 1 – beendet: Verzögerung = 2074ms, Abfrage = 281ms, Laden = 137183ms, Zeilen = 1951

Abschnitt 0 – beendet: Verzögerung = 1109ms, Abfrage = 440ms, Laden = 143795ms, Zeilen = 2049

Gesamtzahl der Zeilen: 4000

Insgesamt verstrichene Zeit: 145346ms

3 Abschnitte:

tanda:wgasior[119]%

tanda:wgasior[119]%java Pload 3

Paralleler Ladetest: 3 Abschnitte (s)

Starten der Zeitgeber.....

Erzeugen der Stränge.....

Verstrichenen Zeit: 1ms

Starten der Stränge.....

Verbinden der Stränge.....

Abschnitt 0 – begonnen: Start = 0, Ende = 333333

Abschnitt 1 – begonnen: Start = 333334, Ende = 666666

Abschnitt 2 – begonnen: Start = 666667, Ende = 999999

Abschnitt 1 – beendet: Verzögerung = 3313ms, Abfrage = 278ms, Laden = 89861ms, Zeilen = 1273

Abschnitt 2 – beendet: Verzögerung = 4096ms, Abfrage = 280ms, Laden = 93512ms, Zeilen = 1324

Abschnitt 0 – beendet: Verzögerung = 2376ms, Abfrage = 434ms, Laden = 98853ms, Zeilen = 1403

Gesamtzahl der Zeilen: 4000

Insgesamt verstrichene Zeit: 101665ms

4 Abschnitte:

tanda:wgasior[120]% java Pload 4

Paralleler Ladetest: 4 Abschnitte (s)

Starten der Zeitgeber.....

Erzeugen der Stränge.....

Verstrichenen Zeit: 1ms

Starten der Stränge.....

Verbinden der Stränge.....

Abschnitt 0 – begonnen: Start = 0, Ende = 249999

Abschnitt 1 – begonnen: Start = 250000, Ende = 499999

Abschnitt 2 – begonnen: Start = 500000, Ende = 749999

Abschnitt 3 – begonnen: Start = 750000, Ende = 999999

Abschnitt 2 – beendet: Verzögerung = 3230ms, Abfrage = 279ms, Laden = 68986ms, Zeilen = 977
Abschnitt 3 – beendet: Verzögerung = 4053ms, Abfrage = 402ms, Laden = 68849ms, Zeilen = 974
Abschnitt 1 – beendet: Verzögerung = 2268ms, Abfrage = 278ms, Laden = 71166ms, Zeilen = 1003
Abschnitt 0 – beendet: Verzögerung = 1398ms, Abfrage = 436ms, Laden = 73751ms, Zeilen = 1046
Gesamtzahl der Zeilen: 4000
Insgesamt verstrichene Zeit: 75588ms

Fünf Abschnitte:

tanda:wgasior[121]% java Pload 5
Paralleler Ladetest: 5 Abschnitte (s)
Starten der Zeitgeber.....
Erzeugen der Stränge.....
Verstrichenen Zeit: 1ms
Starten der Stränge.....
Verbinden der Stränge.....
Abschnitt 0 – begonnen: Start = 0, Ende = 199999
Abschnitt 1 – begonnen: Start = 200000, Ende = 399999
Abschnitt 2 – begonnen: Start = 400000, Ende = 599999
Abschnitt 3 – begonnen: Start = 600000, Ende = 799999
Abschnitt 4 – begonnen: Start = 800000, Ende = 999999
Abschnitt 2 – beendet: Verzögerung = 2869ms, Abfrage = 279ms, Laden = 53344ms, Zeilen = 747
Abschnitt 3 – beendet: Verzögerung = 3874ms, Abfrage = 280ms, Laden = 55950ms, Zeilen = 786
Abschnitt 1 – beendet: Verzögerung = 1996ms, Abfrage = 280ms, Laden = 58800ms, Zeilen = 827
Abschnitt 0 – beendet: Verzögerung = 1119ms, Abfrage = 442ms, Laden = 59976ms, Zeilen = 842
Abschnitt 4 – beendet: Verzögerung = 4640ms, Abfrage = 281ms, Laden = 56906ms, Zeilen = 798
Gesamtzahl der Zeilen: 4000
Insgesamt verstrichene Zeit: 61829ms

6 Abschnitte:

tanda:wgasior[122]% java Pload 6
Paralleler Ladetest: 6 Abschnitte (s)
Starten der Zeitgeber.....
Erzeugen der Stränge.....
Verstrichenen Zeit: 1ms
Starten der Stränge.....
Verbinden der Stränge.....
Abschnitt 0 – begonnen: Start = 0, Ende = 166666
Abschnitt 1 – begonnen: Start = 166667, Ende = 333333
Abschnitt 2 – begonnen: Start = 333334, Ende = 500000

Abschnitt 3 – begonnen: Start = 500001, Ende = 666667

Abschnitt 4 – begonnen: Start = 666668, Ende = 833333

Abschnitt 5 – begonnen: Start = 833334, Ende = 999999

Abschnitt 3 – beendet: Verzögerung = 3804ms, Abfrage = 277ms, Laden = 44795ms, Zeilen = 627

Abschnitt 2 – beendet: Verzögerung = 3018ms, Abfrage = 286ms, Laden = 46287ms, Zeilen = 646

Abschnitt 0 – beendet: Verzögerung = 1308ms, Abfrage = 437ms, Laden = 49973ms, Zeilen = 701

Abschnitt 4 – beendet: Verzögerung = 4594ms, Abfrage = 279ms, Laden = 47290ms, Zeilen = 662

Abschnitt 1 – beendet: Verzögerung = 2234ms, Abfrage = 279ms, Laden = 50069ms, Zeilen = 702

Abschnitt 5 – beendet: Verzögerung = 5375ms, Abfrage = 281ms, Laden = 47479ms, Zeilen = 662

Gesamtzahl der Zeilen: 4000

Insgesamt verstrichene Zeit: 53138ms

7 Abschnitte:

tanda:wgasior[123]% java Pload 7

Paralleler Ladetest: 7 Abschnitte (s)

Starten der Zeitgeber.....

Erzeugen der Stränge.....

Verstrichenen Zeit: 2ms

Starten der Stränge.....

Verbinden der Stränge.....

Abschnitt 0 – begonnen: Start = 0, Ende = 142857

Abschnitt 1 – begonnen: Start = 142858, Ende = 285714

Abschnitt 2 – begonnen: Start = 285715, Ende = 428571

Abschnitt 3 – begonnen: Start = 428572, Ende = 571428

Abschnitt 4 – begonnen: Start = 571429, Ende = 714285

Abschnitt 5 – begonnen: Start = 714286, Ende = 857142

Abschnitt 6 – begonnen: Start = 857143, Ende = 999999

Abschnitt 3 – beendet: Verzögerung = 3568ms, Abfrage = 282ms, Laden = 37341ms, Zeilen = 517

Abschnitt 1 – beendet: Verzögerung = 1957ms, Abfrage = 277ms, Laden = 42125ms, Zeilen = 586

Abschnitt 0 – beendet: Verzögerung = 1117ms, Abfrage = 439ms, Laden = 43081ms, Zeilen = 599

Abschnitt 2 – beendet: Verzögerung = 2781ms, Abfrage = 278ms, Laden = 42408ms, Zeilen = 589

Abschnitt 6 – beendet. Verzögerung = 5998ms, Abfrage = 283ms, Laden = 39964ms, Zeilen = 554

Abschnitt 4 – beendet: Verzögerung = 4341ms, Abfrage = 278ms, Laden = 41839ms, Zeilen = 581

Abschnitt 5 – beendet: Verzögerung = 5128ms, Abfrage = 282ms, Laden = 41499ms, Zeilen = 574

Gesamtzahl der Zeilen: 4000

Insgesamt verstrichene Zeit: 46912ms

8 Abschnitte:

tanda:wgasior[124]% java Pload 8

Paralleler Ladetest: 8 Abschnitte (s)
Starten der Zeitgeber.....
Erzeugen der Stränge.....
Verstrichenen Zeit: 2ms
Starten der Stränge.....
Verbinden der Stränge.....
Abschnitt 0 – begonnen: Start = 0, Ende = 124999
Abschnitt 1 – begonnen: Start = 125000, Ende = 249999
Abschnitt 2 – begonnen: Start = 250000, Ende = 374999
Abschnitt 3 – begonnen: Start = 375000, Ende = 499999
Abschnitt 4 – begonnen: Start = 500000, Ende = 624999
Abschnitt 5 – begonnen: Start = 625000, Ende = 749999
Abschnitt 6 – begonnen: Start = 750000, Ende = 874999
Abschnitt 7 – begonnen: Start = 875000, Ende = 999999
Abschnitt 3 – beendet: Verzögerung = 3718ms, Abfrage = 281ms, Laden = 33849ms, Zeilen = 466
Abschnitt 4 – beendet: Verzögerung = 4486ms, Abfrage = 278ms, Laden = 33419ms, Zeilen = 460
Abschnitt 0 – beendet: Verzögerung = 1290ms, Abfrage = 434ms, Laden = 37526ms, Zeilen = 518
Abschnitt 1 – beendet: Verzögerung = 2158ms, Abfrage = 280ms, Laden = 38303ms, Zeilen = 528
Abschnitt 7 – beendet: Verzögerung = 6921ms, Abfrage = 277ms, Laden = 34587ms, Zeilen = 477
Abschnitt 2 – beendet: Verzögerung = 2943ms, Abfrage = 276ms, Laden = 38879ms, Zeilen = 537
Abschnitt 6 – beendet. Verzögerung = 6145ms, Abfrage = 294ms, Laden = 35982ms, Zeilen = 497
Abschnitt 5 – beendet: Verzögerung = 5274ms, Abfrage = 285ms, Laden = 37516ms, Zeilen = 517
Gesamtzahl der Zeilen: 4000
Insgesamt verstrichene Zeit: 43078ms

9 Abschnitte:
tanda:wgasior[125]% java Pload 9
Paralleler Ladetest: 9 Abschnitte (s)
Starten der Zeitgeber.....
Erzeugen der Stränge.....
Verstrichenen Zeit: 2ms
Starten der Stränge.....
Verbinden der Stränge.....
Abschnitt 0 – begonnen: Start = 0, Ende = 111111
Abschnitt 1 – begonnen: Start = 111112, Ende = 222222
Abschnitt 2 – begonnen: Start = 222223, Ende = 333333
Abschnitt 3 – begonnen: Start = 333334, Ende = 444444
Abschnitt 4 – begonnen: Start = 444445, Ende = 555555
Abschnitt 5 – begonnen: Start = 555556, Ende = 666666

Abschnitt 6 – begonnen: Start = 666667, Ende = 777777

Abschnitt 7 – begonnen: Start = 777778, Ende = 888888

Abschnitt 8 – begonnen: Start = 888889 Ende = 999999

Abschnitt 4 – beendet: Verzögerung = 4282ms, Abfrage = 284ms, Laden = 30014ms, Zeilen = 406

Abschnitt 3 – beendet: Verzögerung = 3524ms, Abfrage = 277ms, Laden = 31543ms, Zeilen = 428

Abschnitt 0 – beendet: Verzögerung = 1117ms, Abfrage = 446ms, Laden = 34725ms, Zeilen = 471

Abschnitt 1 – beendet: Verzögerung = 1932ms, Abfrage = 281ms, Laden = 34593ms, Zeilen = 470

Abschnitt 2 – beendet: Verzögerung = 2757ms, Abfrage = 280ms, Laden = 34124ms, Zeilen = 462

Abschnitt 5 – beendet: Verzögerung = 5172ms, Abfrage = 289ms, Laden = 32478ms, Zeilen = 439

Abschnitt 8 – beendet: Verzögerung = 7635ms, Abfrage = 281ms, Laden = 31267ms, Zeilen = 422

Abschnitt 7 – beendet: Verzögerung = 6865ms, Abfrage = 279ms, Laden = 32822ms, Zeilen = 443

Abschnitt 6 – beendet. Verzögerung = 6034ms, Abfrage = 283ms, Laden = 33880ms, Zeilen = 459

Gesamtzahl der Zeilen: 4000

Insgesamt verstrichene Zeit: 40199ms

10 Abschnitte:

tanda:wgasior[126]% java Pload 10

Paralleler Ladetest: 10 Abschnitte (s)

Starten der Zeitgeber.....

Erzeugen der Stränge.....

Verstrichenen Zeit: 3ms

Starten der Stränge.....

Verbinden der Stränge.....

Abschnitt 0 – begonnen: Start = 0, Ende = 99999

Abschnitt 1 – begonnen: Start = 100000, Ende = 199999

Abschnitt 2 – begonnen: Start = 200000, Ende = 299999

Abschnitt 3 – begonnen: Start = 300000, Ende = 399999

Abschnitt 4 – begonnen: Start = 400000, Ende = 499999

Abschnitt 5 – begonnen: Start = 500000, Ende = 599999

Abschnitt 6 – begonnen: Start = 600000, Ende = 699999

Abschnitt 7 – begonnen: Start = 700000, Ende = 799999

Abschnitt 8 – begonnen: Start = 800000, Ende = 899999

Abschnitt 9 – begonnen: Start = 900000, Ende = 999999

Abschnitt 1 – beendet: Verzögerung = 2054ms, Abfrage = 281ms, Laden = 30277ms, Zeilen = 408

Abschnitt 4 – beendet: Verzögerung = 4544ms, Abfrage = 284ms, Laden = 28206ms, Zeilen = 380

Abschnitt 5 – beendet: Verzögerung = 5452ms, Abfrage = 281ms, Laden = 27309ms, Zeilen = 367

Abschnitt 2 – beendet: Verzögerung = 2818ms, Abfrage = 277ms, Laden = 29958ms, Zeilen = 404

Abschnitt 0 – beendet: Verzögerung = 1116ms, Abfrage = 441ms, Laden = 32200ms, Zeilen = 434

Abschnitt 3 – beendet: Verzögerung = 3681ms, Abfrage = 278ms, Laden = 31292ms, Zeilen = 423

Abschnitt 6 – beendet. Verzögerung = 6243ms, Abfrage = 287ms, Laden = 28825ms, Zeilen = 386

Abschnitt 7 – beendet: Verzögerung = 7088ms, Abfrage = 284ms, Laden = 29758ms, Zeilen = 400

Abschnitt 9 – beendet: Verzögerung = 8736ms, Abfrage = 314ms, Laden = 28869ms, Zeilen = 389

Abschnitt 8 – beendet: Verzögerung = 7915ms, Abfrage = 285ms, Laden = 30348ms, Zeilen = 409

Gesamtzahl der Zeilen: 4000

Insgesamt verstrichene Zeit: 38550ms

[0043] Die vorstehenden Beispiele listen elf Tests verschiedener Aufteilungen auf. Das erste Beispiel war das einer Übertragung mit drei Abschnitten über drei Stränge. Die nächsten 10 Beispiele waren aufeinanderfolgende Beispiele mit eins bis zehn Abschnitten für jede Aufteilung. Bei jeder Aufteilung wurden 4000 Datenzeilen verwendet. Jedes Beispiel verwendete auch eine unterschiedliche Anzahl von Knotenverbindungen je nach der Anzahl der bei der Durchführung erforderlichen Aufteilungen.

[0044] Eine Softwareimplementierung der oben beschriebenen Ausführungsform weist eventuell eine Reihe von Computeranweisungen auf, die entweder auf einem konkret fassbaren Gegenstand fixiert sind, wie z. B. einem computerlesbaren Medium, z. B. einer Diskette, einer CD-ROM, einem ROM-Speicher oder auf einer Festplatte, oder die über ein Computersystem übertragbar sind, und zwar über ein Modem oder eine andere Schnittstelleneinrichtung über ein Medium. Das Medium kann entweder ein konkret fassbares Medium sein, einschließlich optischer oder analoger Kommunikationsleitungen, ohne darauf beschränkt zu sein, oder kann mit drahtlosen Techniken implementiert sein, einschließlich von Mikrowellen, Infrarot- oder anderen Übertragungstechniken, ohne hierauf beschränkt zu sein. Es kann auch das Internet sein. Die Serie von Computeranweisungen verkörpert alle oder einen Teil der Funktionen, die hier unter Bezug auf die Erfindung vorstehend beschrieben wurden. Fachleute auf diesem Gebiet erkennen, daß solche Computeranweisungen in einer Anzahl von Programmiersprachen für die Verwendung mit vielen Computerarchitekturen oder Betriebssystemen geschrieben werden können. Weiterhin können solche Anweisungen unter Verwendung irgendeiner derzeitigen oder künftigen Speichertechnologie gespeichert werden, was Halbleiter-, magnetische, optische oder sonstige Speichereinrichtungen einschließt, ohne hierauf beschränkt zu sein, oder sie können unter Verwendung irgendeiner derzeitigen oder künftigen Kommunikationstechnologie übertragen werden, einschließlich optischer, infraroter, Mikrowellen- oder anderer Übertragungstechnologien, ohne jedoch hierauf beschränkt zu sein. Man kann sich vorstellen, daß solch ein Computerprogrammprodukt als getrenntes Medium mit einer zugehörigen gedruckten oder elektronischen Dokumentation, z. B. einer komprimierten, verpackten Software, vorinstalliert in einem Computersystem, beispielsweise auf einem System ROM oder einer Festplatte verteilt werden kann, oder von einem Server oder einem elektronischen schwarzen Brett über ein Netzwerk, beispielsweise das Internet oder das World Wide Web verteilt werden kann.

[0045] Auch wenn eine beispielhafte Ausführungsform der Erfindung offenbart worden ist, liegt es für Fachleute auf der Hand, daß verschiedene Veränderungen und Modifikationen vorgenommen werden können, die einige der Vorteile der Erfindung erreichen, ohne vom Schutzumfang der Erfindung abzuweichen. Es ist beispielsweise für einigermaßen erfahrene Fachleute offensichtlich, daß, auch wenn die Beschreibung auf ein bestimmtes Hardwaresystem und Betriebssystem gerichtet war, andere Hardware- und andere Betriebssystemsoftware auf dieselbe Weise wie beschrieben verwendet werden könnte. Andere Aspekte, wie z. B. die speziellen Anweisungen, die verwendet wurden, um eine bestimmte Funktion zu erfüllen, ebenso wie andere Modifikationen des erfinderischen Konzepts sollen durch den Schutzumfang der Erfindung abgedeckt sein.

Patentansprüche

1. Verteiltes Computersystem (**100**) für das Übertragen eines Datensatzes (**310**) zwischen einem ersten Computerspeicherort und einem zweiten Computerspeicherort, wobei das System aufweist:
eine Mehrzahl von Kommunikationsknoten (**126**, **128**), welche die ersten und zweiten Computerspeicherorte miteinander verbinden, wobei jeder aus der Mehrzahl von Kommunikationsknoten als ein Kommunikationsstrang dient,
einen Datenübertragungscontroller (**132**), der mit dem ersten oder zweiten Computerspeicherort und der Mehrzahl von Kommunikationsknoten verbunden ist, um eine Anzahl von Kommunikationssträngen bzw. -pfaden auszuwählen, die als Datenübertragungsglieder zwischen dem ersten und zweiten Computerspeicherort dienen,
einen Datenaufteiler (**134**), der mit dem ersten Computerspeicherort verbunden ist, wobei der Datenaufteiler auf die Übertragungsanforderung für einen Datensatz in der Weise reagiert, daß er den Datensatz in übertragbare Datensatzabschnitte aufteilt, wobei der Datensatz Datenzeilen aufweist, die einen Gesamtbereich haben,

und wobei das Aufteilen des Datensatzes aufweist:
Definieren einer Aufteilungsspalte, um jeden Datensatzabschnitt zu markieren, Auswählen eines Teilbereiches von Datenzeilen, um jeden Datensatzabschnitt zu definieren, und Zuordnen des Bereiches von Datenzeilen zu einem gegebenen Datensatzabschnitt, und wobei der Datenübertragungscontroller die Datensatzabschnitte parallel über die Kommunikationsstränge überträgt.

2. Verteiltes Computersystem nach Anspruch 1, wobei der Datensatz in eine Anzahl von Datensatzabschnitten aufgeteilt ist, die gleich der Anzahl der ausgewählten Kommunikationsstränge bzw. -pfade ist.

3. Verteiltes Computersystem nach Anspruch 1, wobei der Datensatz in eine Anzahl von Datensatzabschnitten aufgeteilt ist, die größer ist als die Anzahl der ausgewählten Kommunikationsstränge.

4. Verteiltes Computersystem nach Anspruch 3, wobei ein erster Satz der Datensatzabschnitte, welcher gleich der Anzahl der ausgewählten Kommunikationsstränge ist, parallel über die Kommunikationsstränge übertragen wird, und wobei etwaige verbleibende Datensatzabschnitte über Kommunikationsstränge übertragen werden, die die Übertragung eines entsprechenden Abschnittes des ersten Satzes von Datensatzabschnitten abgeschlossen haben, so daß einige der Datensatzabschnitte seriell über die parallelen Kommunikationsstränge übertragen werden.

5. Verteiltes Computersystem nach einem der vorstehenden Ansprüche, wobei der Datenübertragungscontroller einen Zugriff auf einen übertragenen Datensatzabschnitt vor dem Abschluß der Übertragung aller Datensatzabschnitte erlaubt.

6. Verteiltes Computersystem nach einem der vorstehenden Ansprüche, wobei der Datenaufteiler eine Aufteilung auf der Basis von Datenzeilen definiert, die verwendet werden, um den Datensatz zu definieren.

7. Verteiltes Computersystem nach einem der vorstehenden Ansprüche, wobei die Aufteilungsspalte verwendet wird, um eine Abschnittsnummer für jede der Datenzeilen zu speichern.

8. Verteiltes Computersystem nach Anspruch 7, wobei die Abschnittsnummern auf Zufallsbasis erzeugt werden.

9. Verteiltes Computersystem nach Anspruch 7 oder 8, wobei die Datenzeilen für einen Abschnitt durch Auswahl eines Bereiches von Abschnittsnummern bestimmt werden.

10. Verteiltes Computersystem nach einem der vorstehenden Ansprüche, wobei das verteilte Computersystem ein Fernnetz (wide area network) aufweist.

11. Verteiltes Computersystem nach einem der vorstehenden Anspruche, wobei das verteilte Computersystem ein Internetsystem aufweist.

12. Verfahren zur Übertragung eines Datensatzes (**310**) zwischen ersten und zweiten Computerspeicherorten in einem verteilten Computersystem (**100**), welches einen ersten Computerspeicherort, einen zweiten Computerspeicherort und eine Mehrzahl von Kommunikationsknoten (**126**, **128**) aufweist, welche die ersten und zweiten Speicherorte miteinander verbinden, wobei das Verfahren aufweist:
Aufteilen des Datensatzes in übertragbare Datensatzabschnitte, wobei der Datensatz Datenzeilen aufweist, die einen Gesamtbereich haben, und wobei das Aufteilen des Datensatzes aufweist:
Definieren einer Abschnittsspalte, um jeden Datensatzabschnitt zu markieren, Auswählen eines Teilbereichs von Datenzeilen, um jeden Datensatzabschnitt zu definieren, und Zuordnen des Bereiches von Datenzeilen zu einem gegebenen Datensatzabschnitt,
Ausbilden einer Mehrzahl von gleichzeitigen Kommunikationssträngen, die aus der Mehrzahl von Kommunikationsknoten ausgewählt werden und Übertragen der Datensatzabschnitte parallel über die Mehrzahl von Kommunikationssträngen zwischen den ersten und zweiten Computerspeicherorten.

13. Verfahren nach Anspruch 12, wobei der Schritt des Ausbildens von Kommunikationssträngen das Auswählen von N Kommunikationssträngen aufweist, wobei N die Zahl der Datensatzabschnitte ist.

14. Verfahren nach Anspruch 12, wobei der Schritt des Ausbildens von Kommunikationssträngen das Auswählen von N Kommunikationssträngen aufweist, wobei N kleiner ist als die Anzahl der Datensatzabschnitte.

15. Verfahren nach Anspruch 14, wobei der Schritt des Übertragens aufweist:
Übertragen von N Datensatzabschnitten über die N Kommunikationsstränge in einer parallelen Übertragung, wobei jeweils ein Datensatzabschnitt einem Kommunikationsstrang zugewiesen ist, sobald jeweils einer der N Kommunikationsstränge verfügbar wird, Zuordnen der verbleibenden Datensatzabschnitte zu den verfügbaren Kommunikationssträngen für eine Übertragung, um auf diese Weise die Übertragungsgeschwindigkeit des aufgeteilten Datensatzes zu optimieren.

16. Verfahren nach einem der Ansprüche 12 bis 15, welches weiterhin das Gewähren eines Zugriffs auf einen vollständig übertragenen Datensatzabschnitt vor der vollständigen Übertragung aller Datensatzabschnitte aufweist.

17. Verfahren nach einem der Ansprüche 12 bis 16, wobei jeder Teilbereich von Datenzeilen für jeden Datensatzabschnitt nicht mit irgendeinem anderen Teilbereich von Datenzeilen für irgendeinen anderen Datensatzabschnitt überlappt.

18. Verfahren nach einem der Ansprüche 12 bis 17, wobei der Schritt des Auswählens eines Teilbereiches das Erzeugen von im wesentlichen gleichmäßig verteilten Werten aufweist, um jeden Teilbereich innerhalb des Gesamtbereiches zu definieren.

19. Verfahren nach einem der Ansprüche 12 bis 18, welches weiterhin das Wiederzusammensetzen der Datensatzabschnitte zu dem Datensatz aufweist.

20. Verfahren nach einem der Ansprüche 12 bis 19, wobei die Abschnittsspalte verwendet wird, um eine Abschnittsnummer für jede der Datenzeilen zu speichern.

21. Verfahren nach Anspruch 20, wobei die Abschnittsnummem auf Zufallsbasis erzeugt werden.

22. Verfahren nach Anspruch 20 oder 21, wobei die Datenzeilen für einen Abschnitt durch Auswählen auf einem Bereich von Abschnittsnummem bestimmt werden.

23. Computerprogrammprodukt für die Verwendung in einem verteilten Computersystem (**100**), welches einen ersten Computerspeicherort, einen zweiten Computerspeicherort und eine Mehrzahl von Kommunikationsknoten (**126**, **128**) aufweist, welche die ersten und zweiten Speicherorte miteinander verbinden, wobei das Computerprogrammprodukt Programmcode für das Übertragen eines Datensatzes zwischen den ersten und zweiten Computerspeicherorten bereitstellt, und aufweist:
Computerprogrammcode zum Aufteilen des Datensatzes in übertragbare Datensatzabschnitte, wobei der Datensatz Datenzeilen aufweist, welche einen Gesamtbereich haben und wobei das Aufteilen des Datensatzes aufweist: Definieren einer Abschnittsspalte, um jeden Datensatzabschnitt zu markieren, Auswählen eines Teilbereiches von Datenzeilen, um jeden Datensatzabschnitt zu definieren, und Zuordnen des Bereiches von Datenzeilen zu einem gegebenen Datensatzabschnitt,
Computerprogrammcode zum Ausbilden einer Mehrzahl von gleichzeitigen Kommunikationssträngen, welche aus der Mehrzahl von Kommunikationsknoten ausgewählt werden, und
Computerprogrammcode zum Übertragen der Datensatzabschnitte parallel über die Mehrzahl von Kommunikationssträngen zwischen den ersten und zweiten Computerspeicherorten.

24. Computerprogrammprodukt nach Anspruch 23, wobei der Computerprogrammcode zum Ausbilden einer Mehrzahl von Kommunikationssträngen N Kommunikationsstränge auswählt, wobei N die Anzahl von Datensatzabschnitten ist.

25. Computerprogrammprodukt nach Anspruch 23, wobei der Computerprogrammcode zum Ausbilden der Mehrzahl von Kommunikationssträngen N Kommunikationspfade auswählt, wobei N kleiner als die Anzahl von Datensatzabschnitten ist.

26. Computerprogrammprodukt nach Anspruch 25, wobei der übertragende Computerprogrammcode aufweist:
Computerprogrammcode zum Übertragen von N Datensatzabschnitten über die N Kommunikationsstränge in einer parallelen Übertragung, wobei jeweils ein Datensatzabschnitt jeweils einem Kommunikationsstrang zugeordnet wird,
Computerprogrammcode zum Zuordnen der verbleibenden Datensatzabschnitte zu den verfügbaren Kommunikationssträngen zur Übertragung, um so die Übertragungsgeschwindigkeit des aufgeteilten Datensatzes zu

optimieren, sobald jeweils einer der N Kommunikationsstränge verfügbar wird.

27. Computerprogrammprodukt nach einem der Ansprüche 23 bis 26, welches weiterhin das Gewähren von Zugriff auf einen vollständig übertragenen Datensatzabschnitt vor dem Abschluß der Übertragung aller Datensatzabschnitte aufweist.

28. Computerprogrammprodukt nach einem der Ansprüche 23 bis 27, wobei jeder Teilbereich von Datenzeilen für jeden Datensatzabschnitt nicht mit irgendeinem anderen Teilbereich von Datenzeilen für irgendeinen anderen Datensatzabschnitt überlappt.

29. Computerprogrammprodukt nach einem der Ansprüche 23 bis 28, wobei der den Teilbereich auswählende Computerprogrammcode Computerprogrammcode zum Erzeugen im wesentlichen gleichmäßig verteilter Werte aufweist, um jeweils einen Teilbereich innerhalb des Gesamtbereiches zu definieren.

30. Computerprogrammprodukt nach einem der Ansprüche 23 bis 29, welches weiterhin Computerprogrammcode zum Wiederzusammensetzen der Datensatzabschnitte zu dem Datensatz aufweist.

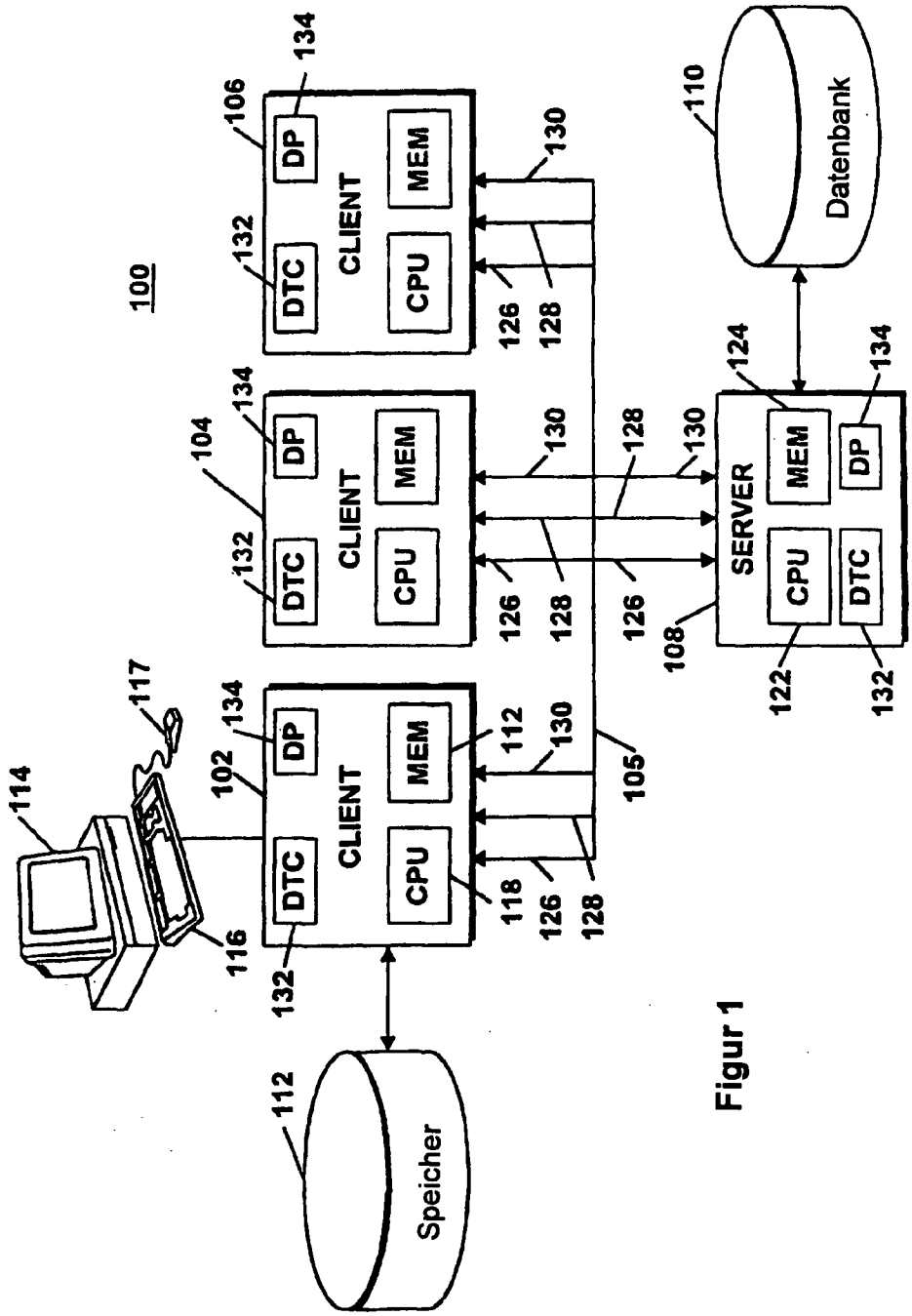
31. Computerprogrammprodukt nach einem der Ansprüche 23 bis 30, wobei die Abschnittsspalte verwendet wird, um eine Abschnittsnummer für jede der Datenzeilen zu speichern.

32. Computerprogrammprodukt nach Anspruch 31, wobei die Abschnittsnummern auf Zufallsbasis erzeugt werden.

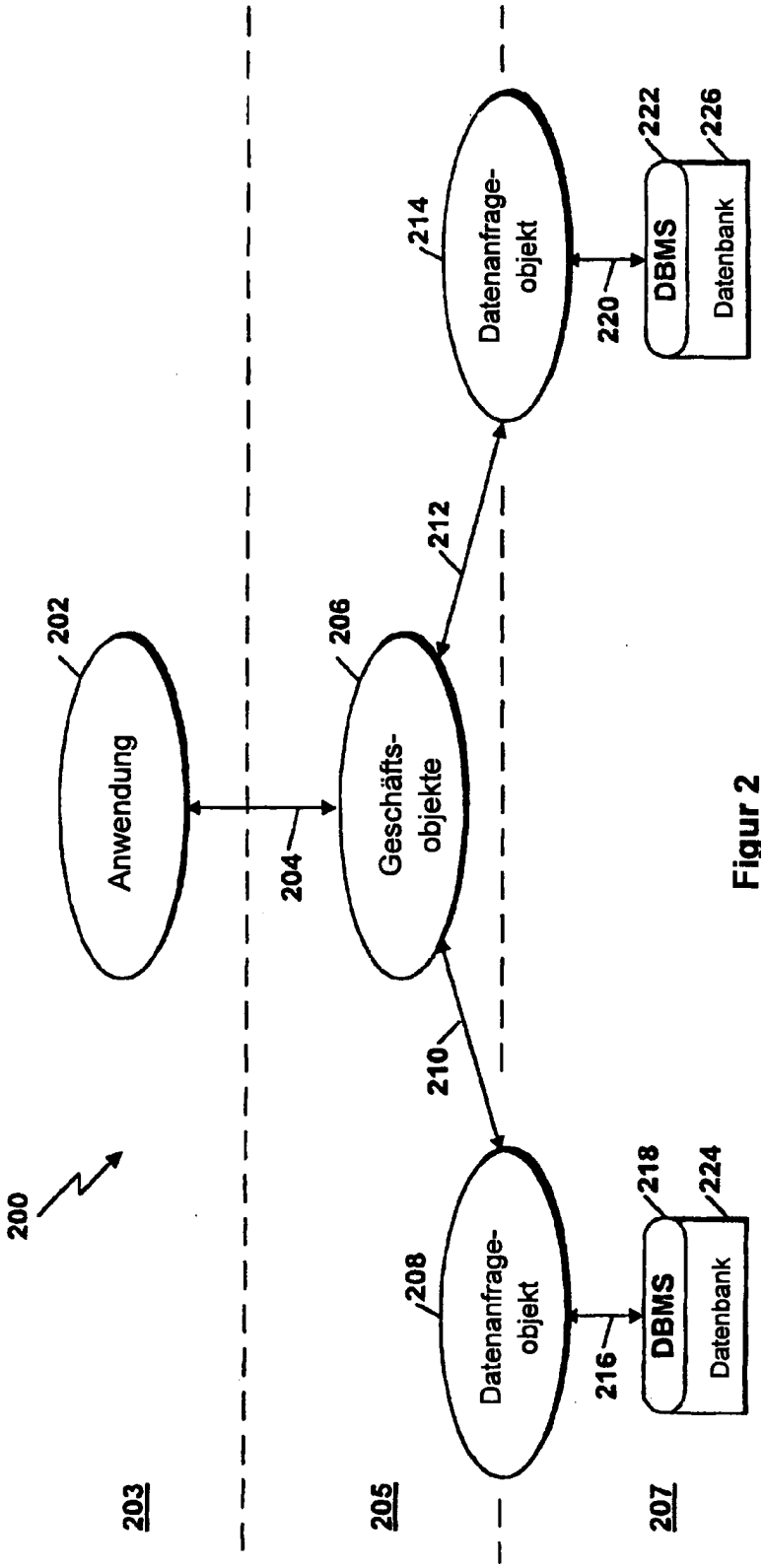
33. Computerprogrammprodukt nach Anspruch 31 oder 32, wobei die Datenzeilen für einen Abschnitt durch Auswählen auf einem Bereich von Abschnittsnummern bestimmt werden.

Es folgen 4 Blatt Zeichnungen

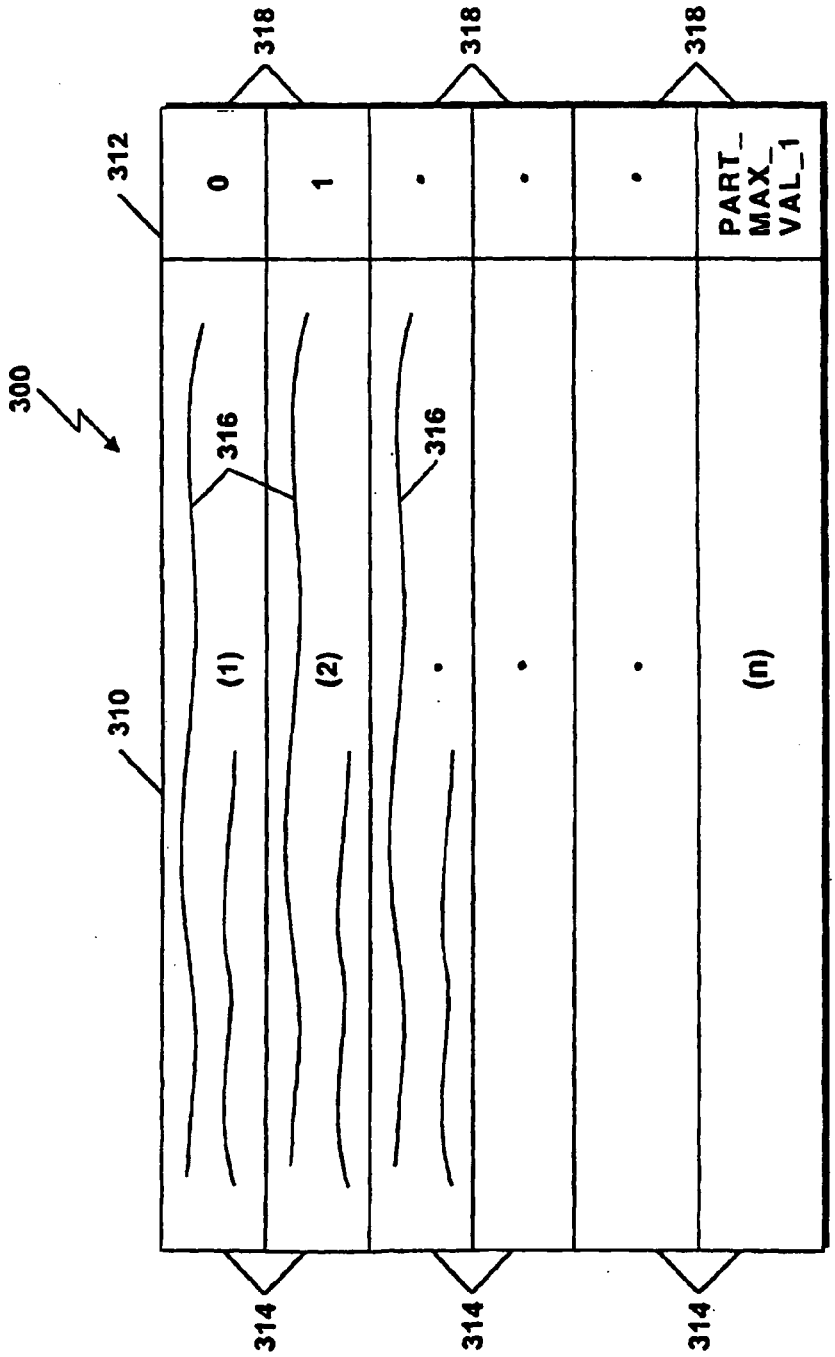
Anhängende Zeichnungen



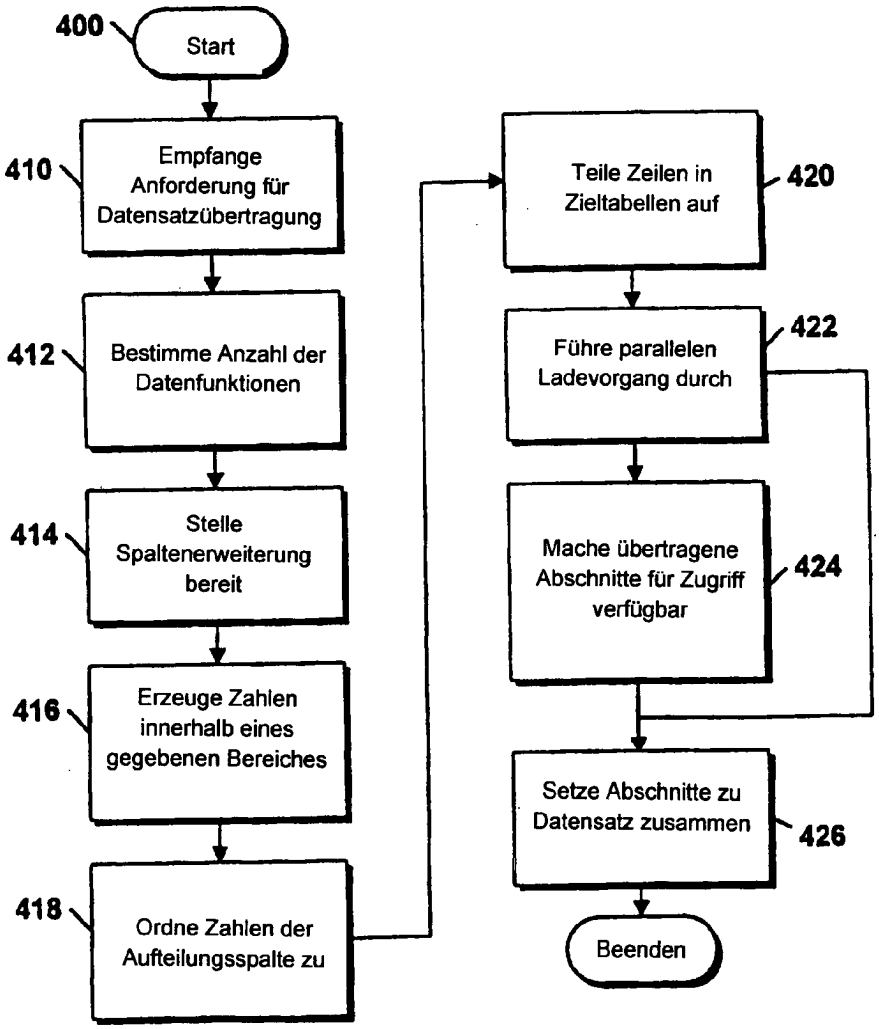
Figur 1


200
202
Anwendung
204
206
Geschäfts-objekte
208
Datenanfrage-objekt
210
212
214
Datenanfrage-objekt
216
218
DBMS
220
222
DBMS
224
Datenbank
226
Datenbank
203
205
207

Figur 2



Figur 3


400
Start
410
Empfange Anforderung für Datensatzübertragung
412
Bestimme Anzahl der Datenfunktionen
414
Stelle Spaltenerweiterung bereit
416
Erzeuge Zahlen innerhalb eines gegebenen Bereiches
418
Ordne Zahlen der Aufteilungsspalte zu
420
Teile Zeilen in Zieltabellen auf
422
Führe parallelen Ladevorgang durch
424
Mache übertragene Abschnitte für Zugriff verfügbar
426
Setze Abschnitte zu Datensatz zusammen
Beenden

Figur 4