



(10) **DE 11 2018 004 402 B4 2022.11.03**

(12)

Patentschrift

(21) Deutsches Aktenzeichen: **11 2018 004 402.5**
 (86) PCT-Aktenzeichen: **PCT/IB2018/057924**
 (87) PCT-Veröffentlichungs-Nr.: **WO 2019/082016**
 (86) PCT-Anmeldetag: **12.10.2018**
 (87) PCT-Veröffentlichungstag: **02.05.2019**
 (43) Veröffentlichungstag der PCT Anmeldung
 in deutscher Übersetzung: **20.05.2020**
 (45) Veröffentlichungstag
 der Patenterteilung: **03.11.2022**

(51) Int Cl.: **G06F 12/02 (2006.01)**
G06F 3/06 (2006.01)
G06F 16/182 (2019.01)

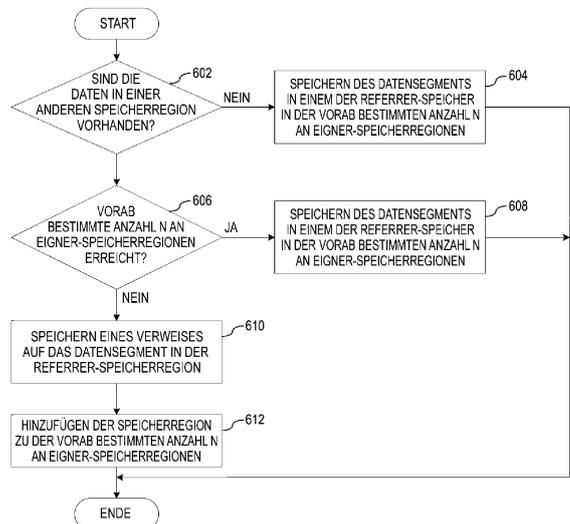
Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

<p>(30) Unionspriorität: 15/793,109 25.10.2017 US</p> <p>(73) Patentinhaber: INTERNATIONAL BUSINESS MACHINES CORPORATION, Armonk, N.Y., US</p> <p>(74) Vertreter: Richardt Patentanwälte PartG mbB, 65185 Wiesbaden, DE</p>	<p>(72) Erfinder: Fischer-Toubol, Jonathan, Tel Aviv, IL; Shatsky, Yosef, Tel Aviv, IL; Halumi, Afief, Tel Aviv, IL; Porat-Stoler, Asaf, Tel Aviv, IL; Marenkov, Sergey, Tel Aviv, IL; Sivan, Tom, Tel Aviv, IL; Cohen, Reut, Tel Aviv, IL; Harnik, Danny, Tel Aviv, IL; Khaitzin, Ety, Tel Aviv, IL</p> <p>(56) Ermittelter Stand der Technik: US 2017 / 0 116 229 A1</p>
--	--

(54) Bezeichnung: **VERBESSERTE LEISTUNG VON VERTEILTER, ORTSBEZOGENER DEDUPLIKATION**

(57) Hauptanspruch: Verfahren in einem Datenverarbeitungssystem (200; 400) für verteilte, ortsbezogene Datenspeicherung, wobei das Verfahren Folgendes umfasst: durch einen Speichermechanismus (406) erfolgreiches Empfangen (502) einer Anfrage von einem Hostsystem (102; 402) zum Schreiben einer Datendatei in eine Referrer-Speicherregion (408) in einer Menge aus Speicherregionen (404a, 404b, 404c, 404d, 404n, 404max) und für jedes Datensegment der Datendatei: in Reaktion darauf, dass ein Vergleich (506) eines Hashwerts für das Datensegment mit anderen Hashwerten für andere in der Referrer-Speicherregion referenzierte gespeicherte Datensegmente anzeigt, dass das Datensegment in der Referrer-Speicherregion nicht vorhanden ist, durch den Speichermechanismus erfolgreiches Ermitteln (602), ob die Daten in einer anderen Speicherregion in der Menge aus Speicherregionen vorhanden sind, in Reaktion darauf, dass das Datensegment in einer anderen Speicherregion in der Menge aus Speicherregionen vorhanden ist, durch den Speichermechanismus erfolgreiches Ermitteln, ob es sich bei der Speicherregion um eine aus einer vorab bestimmten Anzahl N an der Referrer-Speicherregion zugehörigen Eigner-Speicherregionen (410a, 410b, 410c, 410n) handelt, in Reaktion darauf, dass es sich bei der Speicherregion nicht um eine aus der vorab bestimmten Anzahl N an der Referrer-Speicherregion zugehörigen Eigner-Speicherre-

gionen handelt, durch den Speichermechanismus erfolgreiches Ermitteln, ob die Anzahl an Eigner-Speicherregionen ...



Beschreibung

HINTERGRUND

[0001] Die vorliegende Anmeldung betrifft allgemein eine verbesserte Datenverarbeitungsvorrichtung und ein Verfahren und insbesondere Mechanismen zum Verbessern der Leistung von verteilter, ortsbezogener Deduplikation.

[0002] In Speichersystemen handelt es sich bei Deduplikation um einen Prozess des Ersetzens doppelt vorhandener Daten im System durch Zeiger, Verweise oder dergleichen auf eine einzige Instanz der Daten und somit das Verringern der Gesamt-speicheranforderung. Normalerweise handelt es sich bei einem Deduplikationsverweis um einen Metadaten-eintrag, der auf einen Eigner zeigt, bei dem es sich entweder um die Daten selbst oder um die Daten repräsentierende Metadaten handelt. Es gibt zwei grundsätzliche Schemata zum Speichern der Daten: inhaltsbezogene Deduplikation und ortsbezogene Deduplikation. Bei inhaltsbezogener Deduplikation werden im Speicher abgelegte Daten durch den Fingerabdruck, einen Hashwert oder dergleichen der Daten ermittelt. Bei ortsbezogener Deduplikation werden im Speicher abgelegte Daten durch den Speicherort der Daten im Benutzeradressraum (volume-offset) ermittelt.

[0003] Ein zentraler Vorteil ortsbezogener Deduplikation ist die Lokalität der Daten beim Durchführen umfangreicher Lesevorgänge oder sequenzieller Lesevorgänge. Ortsbezogene Deduplikation erfordert weniger Festplatteneingaben/-ausgaben (E/As). Andererseits weist inhaltsbasierte Deduplikation bei wenigen E/As im System einen besseren Ausgleich von Ressourcen auf. Da Deduplikation eine beträchtliche Menge an Direktzugriffsspeicher (RAM) erfordert, unterteilen einige Implementierungen zur Vereinfachung von Verwaltung und/oder Umlagerung die Metadaten in Regionen (Raumunterteilung), beispielsweise in eine Eignerregion und eine Referrer-Region.

[0004] Die US 2017 / 0 116 229 A1 betrifft ein Verfahren zur Optimierung einer Deduplizierung in einer Computerspeicherumgebung durch einen Prozessor. Das Verfahren umfasst ein intelligentes Bilden von Verbindungen zwischen Datenbereichen auf der Grundlage aktueller Popularitätsstatistiken, die eine Anzahl von Malen beinhalten, in denen ein bestimmter der Datenbereiche ein Ziel für eine potentielle Verbindung mit einem anderen der Datenbereiche war.

KURZDARSTELLUNG

[0005] Diese Kurzdarstellung wird bereitgestellt, um in vereinfachter Form eine Auswahl von Konzepten

vorzustellen, die nachstehend in der ausführlichen Beschreibung eingehender beschrieben werden. Diese Kurzdarstellung ist nicht dafür vorgesehen, Schlüsselfaktoren oder Wesensmerkmale des beanspruchten Gegenstands zu identifizieren, und ist auch nicht dafür vorgesehen, den Umfang des beanspruchten Gegenstands einzuschränken.

[0006] In einer veranschaulichenden Ausführungsform wird ein Verfahren in einem Datenverarbeitungssystem für verteilte ortsbezogener Datenspeicherung bereitgestellt. Die veranschaulichende Ausführungsform empfängt von einem Hostsystem eine Anfrage zum Schreiben einer Datendatei in eine Referrer-Speicherregion in einer Menge aus Speicherregionen. Für jedes Datensegment der Datendatei: ermittelt die veranschaulichende Ausführungsform, ob die Daten in einer anderen Speicherregion in der Menge aus Speicherregionen vorhanden sind, in Reaktion darauf, dass ein Vergleich eines Hashwerts für das Datensegment mit anderen Hashwerten für andere gespeicherte Datensegmente, die in der Referrer-Speicherregion referenziert werden, anzeigt, dass das Datensegment in der Referrer-Speicherregion nicht vorhanden ist. In Reaktion darauf, dass das Datensegment in einer anderen Speicherregion in der Menge aus Speicherregionen vorhanden ist, ermittelt die veranschaulichende Ausführungsform, ob es sich bei der Speicherregion um eine einer vorab bestimmten Anzahl N an Eigner-Speicherregionen handelt, die der Referrer-Speicherregion zugehörig sind. In Reaktion darauf, dass es sich bei der Speicherregion nicht um eine der vorab bestimmten Anzahl N an Eigner-Speicherregionen handelt, die der Referrer-Speicherregion zugehörig sind, ermittelt die veranschaulichende Ausführungsform, ob die Anzahl an Eigner-Speicherregionen in der vorab bestimmten Anzahl N an Eigner-Speicherregionen erreicht wurde. In Reaktion darauf, dass die vorab bestimmte Anzahl N an Eigner-Speicherregionen nicht erreicht wurde, speichert die veranschaulichende Ausführungsform einen Verweis auf das Datensegment in der Referrer-Speicherregion.

[0007] In weiteren veranschaulichenden Ausführungsformen wird ein Computerprogrammprodukt bereitgestellt, das ein durch einen Computer verwendbares oder lesbares Medium mit einem computerlesbaren Programm aufweist. Das computerlesbare Programm veranlasst bei Ausführung auf einer Datenverarbeitungseinheit die Datenverarbeitungseinheit, verschiedene der sowie Kombinationen der vorstehend in Bezug auf die veranschaulichende Ausführungsform des Verfahrens umrissenen Arbeitsschritte durchzuführen.

[0008] In einer weiteren Ausführungsform wird ein System bzw. eine Vorrichtung bereitgestellt. Das System bzw. die Vorrichtung kann einen oder meh-

rere Prozessoren und einen mit den einen oder den mehreren Prozessoren verbundenen Arbeitsspeicher umfassen. Der Arbeitsspeicher kann Anweisungen aufweisen, die bei Ausführung durch den einen oder die mehreren Prozessoren den einen oder die mehreren Prozessoren veranlassen, verschiedene der sowie Kombinationen der vorstehend in Bezug auf die veranschaulichende Ausführungsform des Verfahrens umrissenen Arbeitsschritte durchzuführen.

[0009] Diese und weitere Merkmale und Vorteile der vorliegenden Erfindung werden in der nachfolgenden ausführlichen Beschreibung der beispielhaften Ausführungsformen der vorliegenden Erfindung beschrieben bzw. für den Fachmann vor deren Hintergrund ersichtlich.

Figurenliste

[0010] Die Erfindung sowie eine bevorzugte Verwendungsform und weitere Zielsetzungen und Vorteile werden am besten aus einer Bezugnahme auf die nachfolgende ausführliche Beschreibung veranschaulichender Ausführungsformen zusammen mit den begleitenden Zeichnungen ersichtlich, wobei:

Fig. 1 einen Schreibvorgang einer 40-KB-Datei als fünf 8-KB-Segmente in acht verschiedene Eignerregionen zeigt, was gemäß einer veranschaulichenden Ausführungsform zu vielen Arten von Verwaltungsdaten führt;

Fig. 2 ein beispielhaftes Schaubild eines verteilten Datenverarbeitungssystems ist, in dem Aspekte der veranschaulichenden Ausführungsformen implementiert werden können;

Fig. 3 ein beispielhaftes Blockschaubild einer Datenverarbeitungseinheit ist, in der Aspekte der veranschaulichenden Ausführungsformen implementiert werden können;

Fig. 4 ein Funktionsblockdiagramm von Speichermechanismen zeigt, welche gemäß einer veranschaulichenden Ausführungsform die Leistung verteilter, ortsbezogener Deduplikation mit minimaler Auswirkung auf die Deduplikationseffizienz verbessern;

Fig. 5 ein Flussdiagramm der Arbeitsschritte zeigt, die von einem Speichermechanismus durchgeführt werden, der gemäß einer veranschaulichenden Ausführungsform die Leistung verteilter, ortsbezogener Deduplikation mit minimaler Auswirkung auf die Deduplikationseffizienz verbessert; und

Fig. 6 ein Flussdiagramm der Arbeitsschritte zeigt, die gemäß einer veranschaulichenden Ausführungsform von einem Speichermechanismus durchgeführt werden, wenn Deduplikation unter einer Teilmenge aus Speicherregio-

nen, d.h. den Referrer-Speicherregionen und der einen oder den mehreren zugehörigen Eigner-Speicherregionen, umgesetzt werden soll.

AUSFÜHRLICHE BESCHREIBUNG

[0011] Wie vorstehend erwähnt, gibt es zwei grundsätzliche Schemata zum Speichern der Daten: inhaltsbasierte Deduplikation und ortsbezogene Deduplikation. Bei inhaltsbasierter Deduplikation werden im Speicher abgelegte Daten durch den Fingerabdruck, einen Hashwert oder dergleichen der Daten ermittelt. Bei ortsbezogener Deduplikation werden im Speicher abgelegte Daten durch den Speicherort der Daten im Benutzeradressraum (volume-offset) ermittelt. Um ortsbezogene Deduplikation in einem auf Metadatenregionen beruhenden System als ein Beispiel zu nehmen, wird bei einer vielfachen Deduplikation, bei der die Daten (oder die Eignermetadaten) über den gesamten Speicher verteilt sind, ein Lokaltätsvorteil grundsätzlich beeinträchtigt. Wenn beispielsweise eine 64-KB-Datei als acht 8-KB-Segmente gespeichert ist, kann ein Lesevorgang der 64-KB-Datei Zugreifen auf acht unterschiedliche Eignerregionen erfordern, was gemäß einer veranschaulichenden Ausführungsform zu vielen Arten von Verwaltungsdaten führen kann, wie in **Fig. 1** veranschaulicht wird.

[0012] Das heißt, wie in **Fig. 1** gezeigt wird, führt ein Hostsystem 102 einen Lesevorgang an einer Datei aus der Speicherregion 108 durch. In der Speicherregion 108 werden der Datei zugehörige Metadaten gelesen und es werden fünf verschiedene Eigner-Speicherregionen identifiziert, welche zumindest einen Abschnitt der Datei aufweisen, nämlich die Speicherregionen 104, 106, 110, 112 und 114. Somit führt die Speicherregion 108 einen Lesevorgang für den in der Speicherregion 104 befindlichen Abschnitt der Datei aus, führt die Speicherregion 108 einen Lesevorgang für den in der Speicherregion 106 befindlichen Abschnitt der Datei aus, führt die Speicherregion 108 einen Lesevorgang für den in der Speicherregion 110 befindlichen Abschnitt der Datei aus, führt die Speicherregion 108 einen Lesevorgang für den in der Speicherregion 112 befindlichen Abschnitt der Datei aus und führt die Speicherregion 108 einen Lesevorgang für den in der Speicherregion 114 befindlichen Abschnitt der Datei aus. Auf Grundlage dieser Lesevorgänge empfängt die Speicherregion 108 von der Speicherregion 104 eine Antwort H1 für den in der Speicherregion 104 befindlichen Abschnitt der Datei, empfängt die Speicherregion 108 von der Speicherregion 106 eine Antwort H2 für den in der Speicherregion 106 befindlichen Abschnitt der Datei, empfängt die Speicherregion 108 von der Speicherregion 110 eine Antwort H3 für den in der Speicherregion 110 befindlichen Abschnitt der Datei, empfängt die Speicherregion 108 von der Speicherregion 112 eine Antwort H4 für den in der

Speicherregion 112 befindlichen Abschnitt der Datei und empfängt die Speicherregion 108 von der Speicherregion 114 eine Antwort H5 für den in der Speicherregion 114 befindlichen Abschnitt der Datei. Nach Empfangen der Antworten H1, H2, H3, H4 und H5 antwortet die Speicherregion 108 an das Hostsystem 102 mit einer Antwort, die H1, H2, H3, H4 und H5 enthält. Aufgrund der auf fünf verschiedene Eigner-Speicherregionen verteilten Speicherung der Segmente der Datei beinhaltet somit ein Lesevorgang an einer Speicherregion (Speicherregion 108) mehr Lesevorgänge aus anderen Eigner-Speicherregionen, mehr Kommunikation zwischen Speicherregionen im Cluster und Einlagern von Metadaten zwischen Speicherregionen, wenn nicht alle Metadaten in der ursprünglich gelesenen Speicherregion enthalten sind. Diese Verwaltungsdaten führen mit jedem dieser zusätzlichen Lesevorgänge zu einer schweren Leistungsbeeinträchtigung des Systems.

[0013] Entsprechend stellen die veranschaulichenden Ausführungsformen Mechanismen zum Verbessern der Leistung verteilter, ortsbezogener Deduplikation mit minimaler Auswirkung auf die Deduplikationseffizienz bereit. Die veranschaulichenden Ausführungsformen erstellen in intelligenter Weise Duplikationsverbindungen zwischen verschiedenen Referrer- und Eigner-Speicherregionen durch Anwenden dynamischer Verwaltungslogik auf eine Auswahl von Eigner-Speicherregionen innerhalb einer Referrer-Speicherregion. Dies wird durch zwei grundsätzliche Mechanismen erreicht. Der erste Mechanismus begrenzt eine Anzahl an Eigner-Speicherregionen, die mit einem gegebenen Referrer verknüpft werden können (bis zu einer vorab bestimmten Anzahl N an Eigner-Speicherregionen), in einer der folgenden Weisen:

- Unter Verwendung einer „Warteschlangenabarbeitung“-Richtlinie (first come, first served policy), die ein Speichern von Daten in der vorab bestimmten Anzahl N an Eigner-Speicherregionen ermöglicht.
- Unter Verwendung einer „Nutzungshäufigkeit“-Richtlinie, die ein Speichern von Daten auf Grundlage einer Nutzungshäufigkeit einer Eigner-Speicherregion in der Referrer-Speicherregion ermöglicht.
 - Die Nutzungshäufigkeit einer Eigner-Speicherregion kann innerhalb eines einzigen Schreibvorgangs bestimmt werden.
 - Die Nutzungshäufigkeit einer Eigner-Speicherregion kann über mehrere Schreibvorgänge hinweg bestimmt werden.
- Unter Verwendung einer hybriden Richtlinie, die ermöglicht, dass Daten zunächst nach „Warteschlangenabarbeitung“ gespeichert werden, um in einer zweiten vorab bestimmten Anzahl

M von Eigner-Speicherregionen mit $M < N$ gespeichert zu werden, und dann zu ermöglichen, dass Daten auf Grundlage einer Nutzungshäufigkeit der Eigner-Speicherregion in der Referrer-Speicherregion nur bis zu der vorab bestimmten Anzahl N an Eigner-Speicherregionen gespeichert werden.

Der zweite Mechanismus ersetzt eine weniger häufig genutzte Eigner-Speicherregion durch eine häufiger genutzte Eigner-Speicherregion derart, dass eine weniger häufig genutzte Eigner-Speicherregion als bevorzugte Eigner-Speicherregion deaktiviert wird, wenn eine vorteilhaftere Eigner-Speicherregion identifiziert wird.

[0014] Bevor die verschiedenen Aspekte der veranschaulichenden Ausführungsform behandelt werden, ist zunächst zu beachten, dass in der gesamten Beschreibung die Bezeichnung „Mechanismus“ verwendet wird, um Elemente der vorliegenden Erfindung zu bezeichnen, die verschiedene Arbeitsschritte, Funktionen und dergleichen ausführen. Bei einem „Mechanismus“ kann es sich gemäß der Verwendung der Bezeichnung vorliegend um eine Implementierung der Funktionen oder Aspekte der veranschaulichenden Ausführungsformen in Form einer Vorrichtung, einer Prozedur oder eines Computerprogrammprodukts handeln. Im Falle einer Prozedur wird die Prozedur durch eine oder mehrere Einheiten, Vorrichtungen, Computer, Datenverarbeitungssysteme oder dergleichen implementiert. Im Falle eines Computerprogrammprodukts wird die Logik, welche durch Computercode oder in oder auf dem Computerprogrammprodukt enthaltene Anweisungen dargestellt wird, durch eine oder mehrere Hardware-Einheiten ausgeführt, um die Funktionalität zu implementieren oder die Arbeitsschritte durchzuführen, die zu dem bestimmten „Mechanismus“ gehören. Die vorliegend beschriebenen Mechanismen können somit als spezialisierte Hardware, auf Universal-Hardware laufende Software, Software-Anweisungen, die derart auf einem Medium gespeichert sind, dass die Anweisungen durch spezielle oder Universal-Hardware ohne Weiteres ausführbar sind, eine Prozedur oder ein Verfahren zum Ausführen der Funktionen oder eine Kombination beliebiger der vorstehend Genannten implementiert sein.

[0015] Die vorliegende Beschreibung und die Ansprüche können in Bezug auf bestimmte Merkmale und Elemente der veranschaulichenden Ausführungsformen die Ausdrücke „ein/e“, „mindestens eine/r/s von“ und „ein/e oder mehrere“ verwenden. Es ist zu beachten, dass diese Ausdrücke und Formulierungen anzeigen sollen, dass von dem bestimmten Merkmal oder Element mindestens eines in der jeweiligen veranschaulichenden Ausführungsform vorhanden ist, dass jedoch auch mehr als

eines vorhanden sein kann. Das heißt, diese Ausdrücke/Formulierungen sollen die Beschreibung oder die Ansprüche nicht dahingehend einschränken, dass nur ein einziges Merkmal/Element vorhanden ist oder dass zwingend mehrere solche Merkmale/Elemente vorhanden sind. Im Gegenteil erfordern diese Ausdrücke/Formulierungen lediglich mindestens ein einziges Merkmal/Element, wobei im Umfang der Beschreibung und der Ansprüche auch die Möglichkeit mehrerer solcher Merkmale/Elemente umfasst ist.

[0016] Darüber hinaus ist zu beachten, dass die Verwendung der Bezeichnung „Engine“, soweit sie vorliegend zur Beschreibung von Ausführungsformen und Merkmalen der Erfindung verwendet wird, keine Einschränkung auf eine bestimmte Implementierung zum Erreichen und/oder Durchführen der Aktionen, Schritte, Prozesse usw. darstellen soll, die der Engine zuschreibbar sind und/oder von dieser durchgeführt werden. Bei einer Engine kann es sich, ohne jedoch hierauf eingeschränkt zu sein, um Software, Hardware und/oder Firmware oder eine beliebige Kombination aus diesen handeln, welche die angegebenen Funktionen ausführt, darunter, ohne jedoch hierauf eingeschränkt zu sein, jegliche Verwendung eines Universal- und/oder Spezialprozessors in Kombination mit geeigneter Software, die in einem maschinenlesbaren Speicher geladen oder gespeichert und von dem Prozessor ausgeführt wird. Ferner dient ein einer bestimmten Engine zugehöriger Name, soweit nicht anders angegeben, lediglich einer vereinfachten Bezugnahme und soll keine Einschränkung auf eine bestimmte Implementierung darstellen. Zudem kann jedwede einer Engine zugeschriebene Funktionalität ebenso von mehreren Engines ausgeführt werden, in die Funktionalität einer anderen Engine der gleichen oder einer anderen Art eingebunden oder mit dieser kombiniert werden oder auf eine oder mehrere Engines unterschiedlicher Konfiguration verteilt werden.

[0017] Zudem ist zu beachten, dass die nachfolgende Beschreibung eine Vielzahl verschiedener Beispiele für verschiedene Elemente der veranschaulichenden Ausführungsformen verwendet, um beispielhafte Implementierungen der veranschaulichenden Ausführungsformen näher zu veranschaulichen und das Verständnis der Mechanismen der veranschaulichenden Ausführungsformen zu erleichtern. Diese Beispiele sind als nicht einschränkend zu verstehen und sind hinsichtlich der verschiedenen Möglichkeiten zum Implementieren der Mechanismen der veranschaulichenden Ausführungsformen nicht abschließend. Ein Fachmann versteht vor dem Hintergrund der vorliegenden Beschreibung, dass viele weitere alternative Implementierungen für diese verschiedenen Elemente existieren, die zusätzlich zu den oder anstelle der vorliegend angegebenen Beispiele verwendet wer-

den können, ohne vom Grundgedanken und Umfang der vorliegenden Erfindung abzuweichen.

[0018] Die veranschaulichenden Ausführungsformen können somit in vielen verschiedenen Arten von Datenverarbeitungsumgebungen verwendet werden. Um einen Kontext für die Beschreibung der konkreten Elemente und Funktionalität der veranschaulichenden Ausführungsformen bereitzustellen, werden die **Fig. 2** und **Fig. 3** nachfolgend als beispielhafte Umgebungen angegeben, in denen Aspekte der veranschaulichenden Ausführungsformen implementiert werden können. Es ist zu beachten, dass die **Fig. 2** und **Fig. 3** lediglich Beispiele sind und keinerlei Einschränkung hinsichtlich der Umgebungen festlegen oder mit sich bringen, in denen Aspekte oder Ausführungsformen der vorliegenden Erfindung implementiert werden können. An den gezeigten Umgebungen können viele Modifikationen vorgenommen werden, ohne vom Grundgedanken und Umfang der vorliegenden Erfindung abzuweichen.

[0019] **Fig. 2** zeigt eine bildliche Darstellung eines beispielhaften verteilten Datenverarbeitungssystems, in dem Aspekte der veranschaulichenden Ausführungsformen implementiert werden können. Das verteilte Datenverarbeitungssystem 200 kann ein Netzwerk aus Computern aufweisen, in dem Aspekte der veranschaulichenden Ausführungsformen implementiert werden können. Das verteilte Datenverarbeitungssystem 200 enthält mindestens ein Netzwerk 202, bei welchem es sich um das zum Bereitstellen von Datenübertragungsverbindungen zwischen verschiedenen innerhalb des verteilten Datenverarbeitungssystems 200 zusammengesetzten Einheiten und Computern verwendete Medium handelt. Das Netzwerk 202 kann Verbindungen wie beispielsweise drahtgebundene oder drahtlose Kommunikationsverbindungen oder Lichtleiterkabel beinhalten.

[0020] In dem abgebildeten Beispiel sind der Server 204 und der Server 206 zusammen mit der Speichereinheit 208 mit dem Netzwerk 202 verbunden. Zudem sind die Clients 210, 212 und 214 ebenfalls mit dem Netzwerk 202 verbunden. Bei diesen Clients 210, 212 und 214 kann es sich beispielsweise um Personal Computer, Netzwerk-Computer oder dergleichen handeln. In dem abgebildeten Beispiel stellt der Server 204 den Clients 210, 212 und 214 Daten wie beispielsweise Bootdateien, Betriebssystemabbilder und Anwendungen bereit. In dem abgebildeten Beispiel weisen die Clients 210, 212 und 214 mit dem Server 204 eine Client-Beziehung auf. Das verteilte Datenverarbeitungssystem 200 kann weitere Server, Clients und andere nicht gezeigte Einheiten beinhalten.

[0021] In dem abgebildeten Beispiel handelt es sich bei dem verteilten Datenverarbeitungssystem 200 um das Internet, wobei das Netzwerk 202 eine weltweite Sammlung aus Netzwerken und Gateways darstellt, welche die „Transmission Control Protocol/Internet Protocol“- (TCP/IP-) Protokollsuite verwenden, um Daten auszutauschen. Im Kern des Internets befindet sich ein Backbone aus Hochgeschwindigkeits-Datenübertragungsleitungen zwischen Hauptknoten oder Hostcomputern, welche aus tausenden gewerblicher, staatlicher, dem Lehrbetrieb zugehöriger und anderer Computersysteme bestehen, die Daten und Nachrichten weiterleiten. Natürlich kann das verteilte Datenverarbeitungssystem 200 auch so implementiert sein, dass es mehrere unterschiedliche Arten von Netzwerken beinhaltet, beispielsweise ein Intranet, ein lokales Netzwerk, ein Weitverkehrsnetz (WAN) oder dergleichen. Wie vorstehend erwähnt, soll **Fig. 2** ein Beispiel und nicht eine architektonische Einschränkung für verschiedene Ausführungsformen der vorliegenden Erfindung darstellen, weshalb die in **Fig. 2** gezeigten konkreten Elemente hinsichtlich der Umgebungen, in denen die veranschaulichenden Ausführungsformen der vorliegenden Erfindung implementiert werden können, nicht als einschränkend zu betrachten sind.

[0022] Wie in **Fig. 2** gezeigt wird, können eine oder mehrere der Datenverarbeitungseinheiten, z.B. die Server 204, speziell so konfiguriert sein, dass sie einen Mechanismus zum Verbessern der Leistung von verteilter, ortsbezogener Deduplikation mit minimaler Einschränkung der Deduplikationseffizienz implementieren. Das Konfigurieren der Datenverarbeitungseinheit kann das Bereitstellen von anwendungsspezifischer Hardware, Firmware oder dergleichen umfassen, um die Durchführung der Arbeitsschritte und Erzeugung der Ergebnisse zu ermöglichen, die vorliegend in Bezug auf die veranschaulichenden Ausführungsformen beschrieben werden. Das Konfigurieren der Datenverarbeitungseinheit kann zudem oder alternativ das Bereitstellen von in einer oder mehreren Speichereinheiten gespeicherten und in den Arbeitsspeicher einer Datenverarbeitungseinheit wie beispielsweise des Servers 204 geladenen Software-Anwendungen umfassen, um einen oder mehrere Hardware-Prozessoren der Datenverarbeitungseinheit zu veranlassen, die Software-Anwendungen auszuführen, welche die Prozessoren konfigurieren, um die Arbeitsschritte durchzuführen und die Ergebnisse zu erzeugen, die vorliegend in Bezug auf die veranschaulichenden Ausführungsformen beschrieben werden. Darüber hinaus kann eine beliebige Kombination aus anwendungsspezifischer Hardware, Firmware, auf Hardware ausgeführter Software oder dergleichen verwendet werden, ohne vom Grundgedanken und Umfang der veranschaulichenden Ausführungsformen abzuweichen.

[0023] Es ist zu beachten, dass, sobald die Datenverarbeitungseinheit in einer solchen Weise konfiguriert ist, aus der Datenverarbeitungseinheit eine spezialisierte Datenverarbeitungseinheit wird, die speziell dafür konfiguriert ist, die Mechanismen der veranschaulichenden Ausführungsformen zu implementieren, und keine Universal-Datenverarbeitungseinheit mehr darstellt. Darüber hinaus verbessert, wie nachstehend beschrieben, die Implementierung der Mechanismen der veranschaulichenden Ausführungsformen die Funktionalität der Datenverarbeitungseinheit und stellt ein nützliches und konkretes Ergebnis bereit, welches die Verbesserung der Leistung verteilter, ortsbezogener Deduplikation mit minimaler Auswirkung auf die Deduplikationseffizienz ermöglicht.

[0024] Wie vorstehend erwähnt, verwenden die Mechanismen der veranschaulichenden Ausführungsformen speziell konfigurierte Datenverarbeitungseinheiten oder Datenverarbeitungssysteme, um die Arbeitsschritte zum Verbessern der Leistung verteilter, ortsbezogener Deduplikation mit minimaler Auswirkung auf die Deduplikationseffizienz durchzuführen. Diese Datenverarbeitungseinheiten oder Datenverarbeitungssysteme können verschiedene Hardware-Elemente umfassen, die entweder durch Hardware-Konfiguration, Software-Konfiguration oder eine Kombination aus Hardware- und Software-Konfiguration so konfiguriert sind, dass sie eines oder mehrere der vorliegend beschriebenen Systeme/Teilsysteme implementieren. **Fig. 3** ist ein Blockschaubild eines beispielhaften Datenverarbeitungssystems, in dem Aspekte der veranschaulichenden Ausführungsformen implementiert werden können. Das Datenverarbeitungssystem 300 ist ein Beispiel für einen Computer, beispielsweise den Server 204 in **Fig. 2**, in dem sich durch einen Computer verwendbarer Code oder Anweisungen, welche die Prozesse und Aspekte der veranschaulichenden Ausführungsformen der vorliegenden Erfindung implementieren, befinden und/oder ausgeführt werden können, um die Durchführung, die Ergebnisse und die Außenwirkungen der vorliegend beschriebenen veranschaulichenden Ausführungsformen zu erreichen.

[0025] In dem gezeigten Beispiel nutzt das Datenverarbeitungssystem 300 eine Hub-Architektur mit einer „North Bridge und Speichercontroller“-Hub (NB/MCH) 302 und einer „South Bridge und Eingabe/Ausgabe- (E/A-) Controller“-Hub (SB/ICH) 304. Eine Verarbeitungseinheit 306, ein Hauptspeicher 308 und ein Grafikprozessor 310 sind mit der NB/MCH 302 verbunden. Der Grafikprozessor 310 kann mit der NB/MCH 302 durch einen Accelerated Graphics Port (AGP) verbunden sein.

[0026] Im gezeigten Beispiel ist der Adapter 312 für das lokale Netzwerk (LAN) mit der SB/ICH 304 ver-

bunden. Ein Audioadapter 316, ein Tastatur- und Mausadapter 320, ein Modem 322, ein Nur-Lese-Speicher (ROM) 324, ein Festplattenlaufwerk (HDD) 326, ein CD-ROM-Laufwerk 330, „Universal Serial Bus“- (USB-) Anschlüsse und weitere Datenübertragungsanschlüsse 332 sowie PCI/PCIe-Einheiten 334 sind mit der SB/ICH 304 durch einen Bus 338 und einen Bus 340 verbunden. Zu PCI/PCIe-Einheiten können beispielsweise Ethernet-Adapter, Add-In-Karten und PC-Karten für Notebook-Computer zählen. PCI verwendet einen Karten-Buscontroller, PCIe hingegen nicht. Beim ROM 324 kann es sich beispielsweise um ein Flash-Basic Input/Output System (BIOS) handeln.

[0027] Das HDD 326 und das CD-ROM-Laufwerk 330 sind mit der SB/ICH 304 durch den Bus 340 verbunden. Das HDD 326 und das CD-ROM-Laufwerk 330 können beispielsweise eine „Integrated Drive Electronics“- (IDE-) oder eine „Serial Advanced Technology Attachment“- (SATA-) Schnittstelle verwenden. Eine „Super I/O“- (SIO-) Einheit 336 kann mit der SB/ICH 304 verbunden sein.

[0028] Auf der Verarbeitungseinheit 306 läuft ein Betriebssystem. Das Betriebssystem stellt Koordination und Steuerung verschiedener Komponenten innerhalb des Datenverarbeitungssystems 300 in **Fig. 3** bereit. Als Client kann es sich bei dem Betriebssystem um ein handelsübliches Betriebssystem wie beispielsweise Microsoft® Windows 7® handeln. Ein objektorientiertes Programmiersystem wie beispielsweise das Java™-Programmiersystem kann zusammen mit dem Betriebssystem laufen und stellt Aufrufe an das Betriebssystem von Java™-Programmen oder auf dem Datenverarbeitungssystem 300 ausgeführten Anwendungen aus bereit.

[0029] Als Server kann es sich bei dem Datenverarbeitungssystem 300 beispielsweise um ein eServer™-, ein System p®-Computersystem, ein prozessorbasiertes Power™-Computersystem oder dergleichen von IBM handeln, das mit dem Betriebssystem Advanced Interactive Executive (AIX®) oder mit dem LINUX®-Betriebssystem läuft. Bei dem Verarbeitungssystem 300 kann es sich um ein symmetrisches Multiprozessor- (SMP-) System handeln, das mehrere Prozessoren in der Verarbeitungseinheit 306 beinhaltet. Alternativ kann ein Einzelprozessorsystem eingesetzt werden.

[0030] Anweisungen für das Betriebssystem, das objektorientierte Programmiersystem und Anwendungen oder Programme befinden sich auf Speichereinheiten wie beispielsweise dem HDD 326 und können zur Ausführung durch die Verarbeitungseinheit 306 in den Hauptspeicher 308 geladen werden. Die Prozesse veranschaulichender Ausführungsformen der vorliegenden Erfindung können durch die Ver-

arbeitungseinheit 306 mittels durch einen Computer verwendbaren Programmcodes durchgeführt werden, der sich in einem Speicher wie beispielsweise dem Hauptspeicher 308, dem ROM 324 oder beispielsweise in einer oder mehreren Peripherieeinheiten 326 und 330 befinden kann.

[0031] Ein Bussystem wie beispielsweise der Bus 338 oder der Bus 340 wie in **Fig. 3** gezeigt kann aus einem oder mehreren Bussen bestehen. Natürlich kann das Bussystem mittels jedweder Art von Kommunikationsgebilde oder -architektur implementiert werden, die eine Übertragung von Daten zwischen verschiedenen an das Gebilde oder die Architektur angebotenen Komponenten oder Einheiten gewährleistet. Eine Kommunikationseinheit wie beispielsweise der Modem 322 oder der Netzwerkadapter 312 aus **Fig. 3** kann eine oder mehrere zum Übertragen und Empfangen von Daten verwendete Einheiten enthalten. Bei einem Speicher kann es sich beispielsweise um den Hauptspeicher 308, den ROM 324 oder einen Cachespeicher handeln, wie er beispielsweise in der NB/MCH 302 in **Fig. 3** vorhanden ist.

[0032] Wie vorstehend erwähnt, können in einigen veranschaulichenden Ausführungsformen die Mechanismen der veranschaulichenden Ausführungsformen als anwendungsspezifische Hardware, Firmware oder dergleichen oder als in einer Speichereinheit wie beispielsweise dem HDD 326 gespeicherte Anwendungssoftware implementiert sein und zur Ausführung durch einen oder mehrere Hardware-Prozessoren wie beispielsweise die Verarbeitungseinheit 306 oder dergleichen in einen Arbeitsspeicher wie beispielsweise den Hauptspeicher 308 geladen werden. Die in **Fig. 3** gezeigte Datenverarbeitungseinheit wird somit speziell konfiguriert, um die Mechanismen der veranschaulichenden Ausführungsformen zu implementieren, und speziell konfiguriert, um die Arbeitsschritte durchzuführen und die Ergebnisse zu erzeugen, die nachfolgend in Bezug auf den Mechanismus zum Verbessern der Leistung von verteilter, ortsbezogener Deduplikation mit minimaler Auswirkung auf die Deduplikationseffizienz beschrieben werden.

[0033] Ein Fachmann versteht, dass die Hardware in den **Fig. 2** und **Fig. 3** abhängig von der Implementierung variieren kann. Zusätzlich zu oder anstelle der in den **Fig. 2** und **Fig. 3** gezeigten Hardware können weitere interne Hardware- oder Peripherieeinheiten wie beispielsweise ein Flashspeicher, gleichwertiger nichtflüchtiger Speicher oder optische Plattenlaufwerke und dergleichen verwendet werden. Zudem können die Prozesse der veranschaulichenden Ausführungsformen auf ein anderes Multiprozessor-Datenverarbeitungssystem als das vorstehend genannte SMP-System angewendet

werden, ohne vom Grundgedanken und Umfang der vorliegenden Erfindung abzuweichen.

[0034] Darüber hinaus kann das Datenverarbeitungssystem 300 in Form beliebiger einer Anzahl verschiedener Datenverarbeitungssysteme vorliegen, darunter Client-Datenverarbeitungseinheiten, Server-Datenverarbeitungseinheiten, ein Tablet-Computer, ein Laptop-Computer, ein Telefon oder eine andere Kommunikationseinheit, ein persönlicher digitaler Assistent (PDA) oder dergleichen. In einigen veranschaulichenden Beispielen kann es sich bei dem Datenverarbeitungssystem 300 um eine tragbare Datenverarbeitungseinheit handeln, die mit Flashspeicher konfiguriert ist, um nichtflüchtigen Speicher zum Speichern von beispielsweise Betriebssystemdateien und/oder nutzergenerierten Daten bereitzustellen. Im Wesentlichen kann es sich bei dem Datenverarbeitungssystem 300 um jedes bekannte oder künftig entwickelte Datenverarbeitungssystem ohne architektonische Einschränkung handeln.

[0035] Fig. 4 zeigt ein Funktionsblockdiagramm von Speichermechanismen, welche gemäß einer veranschaulichenden Ausführungsform die Leistung verteilter, ortsbezogener Deduplikation mit minimaler Auswirkung auf die Deduplikationseffizienz verbessern. Wie vorstehend behandelt, erstellen die Speichermechanismen in intelligenter Weise Duplikationsverbindungen zwischen verschiedenen Referrer- und Eigner-Speicherregionen durch Anwenden dynamischer Verwaltungslogik auf eine Auswahl von Eigner-Speicherregionen innerhalb der Referrer-Speicherregion. Gemäß den veranschaulichenden Ausführungsformen handelt es sich bei einer Referrer-Region um eine Speicherregion, in welche die Daten zunächst geschrieben werden und die Verweise zu weiteren Speicherregionen, welche Daten speichern (d.h. Eigner-Regionen) oder zu der Referrer-Region selbst besitzt, da die Referrer-Speicherregionen ebenfalls Daten speichern können. In einer ersten Ausführungsform begrenzt ein Speichermechanismus 406 eine Anzahl von Eigner-Speicherregionen, die mit einer gegebenen Referrer-Speicherregion verknüpft werden können (bis zu einer vorab bestimmten Anzahl N an Eigner-Speicherregionen), unter Verwendung einer „Warteschlangenabarbeitung“-Richtlinie, die ein Speichern von Daten in der vorab bestimmten Anzahl N an Eigner-Speicherregionen ermöglicht.

[0036] Wenn in dieser ersten Ausführungsform das Hostsystem 402 im Datenverarbeitungssystem 400 Daten in eine Speicherregion schreibt, schreibt/speichert der Speichermechanismus 406 die Daten beispielsweise in 8-KB-Segmenten. Wenn das Hostsystem 402 eine 16-KB-Datendatei in eine Speicherregion schreibt, beispielsweise die Speicherregion 404a in den Speicherregionen

404a-404max, wird somit der beschriebene Speicher zu einer Referrer-Speicherregion 408, wie durch den gestrichelten Kasten angezeigt. Der Speichermechanismus 406 erzeugt einen Hashwert für jedes der beiden 8-KB-Segmente der 16-KB-Datendatei und vergleicht den Hashwert für jedes 8-KB-Datensegment mit Hashwerten für andere in den Speicherregionen 404a-404max referenzierte gespeicherte Datensegmente. Bei dem Speichermechanismus 406 handelt es sich um eine Einheit, die sich im Host 402 oder in einer oder mehreren der Speicherregionen 404a-404max befinden kann oder als eigenständiger Mechanismus vorliegen kann. Falls die Referrer-Speicherregion 408 anzeigt, dass eines oder beide übereinstimmende Datensegmente bereits in der Referrer-Speicherregion 408 oder einer oder mehreren der Eigner-Speicherregionen 410a-410n vorhanden sind, stellt der Speichermechanismus 406, anstatt die 8-KB-Datensegmente zu speichern, fest, dass spätere Verweise auf diese Datensegmente durch die Hashwertvergleiche mit der Referrer-Speicherregion 408 oder einer oder mehreren der Eigner-Speicherregionen 410a-410n identifiziert werden. Falls jedoch der Vergleich mit den Speicherregionen 404a-404max anzeigt, dass eines oder beide der Datensegmente noch nicht in der Referrer-Speicherregion 408 und den zugehörigen Eigner-Speicherregionen 410a-410n vorhanden sind, speichert der Speichermechanismus 406 das noch nicht vorhandene Datensegment in der Referrer-Speicherregion 408.

[0037] Gemäß der „Warteschlangenabarbeitung“-Richtlinie werden zuvor noch nicht gespeicherte Datensegmente in die Referrer-Speicherregion 408 geschrieben. Jedes Mal, wenn auf Daten in der Referrer-Speicherregion 408 und/oder der einen oder den mehreren der Referrer-Speicherregion 4008 zugehörigen Eigner-Speicherregionen 410a-410n zugegriffen wird, lädt jedoch der Speichermechanismus 406 die Speicherregionen, in denen die Daten vorhanden sind. Wenn eine 64-KB-Datendatei acht 8-KB-Datensegmente besitzt, die in acht verschiedenen Eigner-Speicherregionen gespeichert sind, lädt somit der Speichermechanismus 406 alle acht verschiedenen Speicherregionen, welche die 8-KB-Segmente der 64-KB-Datendatei aufweisen. Das Problem wäre noch größer bei einer 256-KB-Datendatei, bei der die 8-KB-Datensegmente auf 32 verschiedene Eigner-Speicherregionen verteilt gespeichert wären. Gemäß den veranschaulichenden Ausführungsformen speichert somit der Speichermechanismus 406 die 8-KB-Datensegmente in lediglich einer vorbestimmten Anzahl N an Eigner-Speicherregionen. Beim Speichern von Daten in dieser verteilten Weise verfolgt somit der Speichermechanismus 406, wo Datensegmente gespeichert werden, mittels eines Verfolgungsmechanismus wie beispielsweise Datenstruktur, Liste, Tabelle oder dergleichen. Wenn Daten in einer

bestimmten Eigner-Speicherregion in der Gruppe der Referrer-Speicherregion 408 und der zugehörigen Eigner-Speicherregionen 410a-410n gespeichert werden, wird somit diese bestimmte Speicherregion der vorab bestimmten Anzahl N an Eigner-Speicherregionen hinzugefügt. Sobald die Anzahl N an Eigner-Speicherregionen erreicht ist, verwendet der Speichermechanismus 406 nur die Eigner-Speicherregionen in der vorab bestimmten Anzahl N an Eigner-Speicherregionen für zukünftige Speicherung von Datenblöcken.

[0038] Falls der Speichermechanismus 406 ein 8-KB-Datensegment schreibt oder liest, das in einer anderen Eigner-Speicherregion als der Referrer-Speicherregion 408 und/oder den Eigner-Speicherregionen 410a-410n in der vorab bestimmten Anzahl N an Eigner-Speicherregionen vorhanden ist, stellt der Speichermechanismus 406 fest, dass es sich bei der Eigner-Speicherregion um keine der Eigner-Speicherregionen in der vorab bestimmten Anzahl N an Eigner-Speicherregionen handelt. In diesem Fall schreibt der Speichermechanismus 406 das 8-KB-Datensegment in die Referrer-Speicherregion 408. Zwar kann dies das doppelt vorhandene 8-KB-Segment innerhalb der Referrer-Speicherregion 408 und den zugehörigen Eigner-Speicherregionen 410a-410n erzeugen, jedoch verringert das Erzeugen des doppelt vorhandenen 8-KB-Segments die für den Speichermechanismus 406 anfallenden Verwaltungsdaten, indem nicht mehr als die vorab bestimmte Anzahl N an Eigner-Speicherregionen geladen wird. Mittels der „Warteschlangenabarbeitung“-Richtlinie verringert somit der Speichermechanismus 406 Verwaltungsdaten im Speicher oder in der Verarbeitung und ist nicht von der Größe der Nutzereingaben/ausgaben abhängig.

[0039] In einer zweiten Ausführungsform verwendet der Speichermechanismus 406 eine „Nutzungshäufigkeit“-Richtlinie, die das Erzeugen von Deduplikation auf Grundlage einer Nutzungshäufigkeit der Referrer-Speicherregion 408 und der durch die Referrer-Speicherregion 408 identifizierten zugehörigen Eigner-Speicherregionen 410a-410n ermöglicht. Wenn das Hostsystem 402 im Datenverarbeitungssystem 400 Daten in einer oder mehreren der Referrer-Speicherregion 408 und der zugehörigen Eigner-Speicherregionen 410a-410n speichert, werden auch hier wieder die Daten beispielsweise in 8-KB-Datensegmenten gespeichert. Wenn das Hostsystem 402 eine 16-KB-Datendatei in die Referrer-Speicherregion 408 schreibt, erzeugt somit der Speichermechanismus 406 einen Hashwert für jedes der beiden 8-KB-Segmente der 16-KB-Datendatei und vergleicht den Hashwert für jedes 8-KB-Datensegment mit den Hashwerten für andere in den Speicherregionen 404a-404max referenzierte gespeicherte Datensegmente. Falls die Referrer-Speicherregion 408 anzeigt, dass eines oder beide

übereinstimmende Datensegmente bereits in der Referrer-Speicherregion 408 oder einer oder mehreren der Eigner-Speicherregionen 410a-410n vorhanden sind, stellt der Speichermechanismus 406, anstatt die 8-KB-Datensegmente zu speichern, fest, dass spätere Verweise auf diese Datensegmente durch die Hashwertvergleiche mit der Referrer-Speicherregion 408 oder einer oder mehreren der Eigner-Speicherregionen 410a-410n identifiziert werden. Falls jedoch der Vergleich mit den Speicherregionen 404a-404max anzeigt, dass eines oder keines der Datensegmente bereits in der Referrer-Speicherregion 408 oder den zugehörigen Eigner-Speicherregionen 410a-410n vorhanden sind, speichert der Speichermechanismus 406 das noch nicht vorhandene Datensegment in der Referrer-Speicherregion 408.

[0040] Gemäß der „Nutzungshäufigkeit“-Richtlinie verfolgt der Speichermechanismus 406 während des Schreibens und Lesens von Datensegmenten aus den Speicherregionen 404a-404max über einen Zähler, eine Datenstruktur oder dergleichen die Nutzungshäufigkeit jeder Eigner-Speicherregion. Wenn der Wert der Nachverfolgung anzeigt, dass eine bestimmte Eigner-Speicherregion oberhalb eines Grenzwerts liegt, wird diese bestimmte Eigner-Speicherregion der vorab bestimmten Anzahl N an Eigner-Speicherregionen hinzugefügt. Die Häufigkeit von Lese- und Schreibvorgängen an die Eigner-Speicherregionen in der vorab bestimmten Anzahl N an Eigner-Speicherregionen zeigt somit an, dass diese Eigner-Regionen diejenigen sein sollten, an die neue Schreibvorgänge gerichtet werden. Auch hier muss der Speichermechanismus 406 jedes Mal, wenn auf Daten in der Referrer-Speicherregion 408 und/oder den zugehörigen Eigner-Speicherregionen 410a-410n zugegriffen wird, diese Eigner-Speicherregionen dort laden, wo die Daten vorhanden sind. Somit identifiziert gemäß den veranschaulichenden Ausführungsformen der Speichermechanismus 406 die am meisten genutzten Eigner-Speicherregionen, fügt diese Eigner-Speicherregionen der vorab bestimmten Anzahl N an Eigner-Speicherregionen hinzu und nutzt diese Eigner-Speicherregionen in der vorab bestimmten Anzahl N an Eigner-Speicherregionen für zukünftige Speicherung von Datenblöcken.

[0041] Falls der Speichermechanismus 406 ein 8-KB-Datensegment schreibt oder liest, das in einer anderen Eigner-Speicherregion als der Referrer-Speicherregion 408 und/oder den Eigner-Speicherregionen 410a-410n in der vorab bestimmten Anzahl N an Eigner-Speicherregionen vorhanden ist, stellt der Speichermechanismus 406 fest, dass es sich bei der Eigner-Speicherregion um keine der Eigner-Speicherregionen in der vorab bestimmten Anzahl N an Eigner-Speicherregionen handelt. In diesem Fall schreibt der Speichermechanismus 406 das 8-KB-

Datensegment in die Referrer-Speicherregion 408. Zwar kann dies das doppelt vorhandene 8-KB-Segment innerhalb der Referrer-Speicherregion 408 und den zugehörigen Eigner-Speicherregionen 410a-4104n erzeugen, jedoch verringert das Erzeugen des doppelt vorhandenen 8KB-Segments die für den Speichermechanismus 406 anfallenden Verwaltungsdaten, indem nicht mehr als die vorab bestimmte Anzahl N an Eigner-Speicherregionen geladen wird. Mittels der „Nutzungshäufigkeit“-Richtlinie verringert somit der Speichermechanismus 406 Verwaltungsdaten im Speicher oder in der Verarbeitung und ist nicht von der Größe der Nutzereingaben/ausgaben abhängig.

[0042] In einer dritten Ausführungsform nutzt der Speichermechanismus 406 eine Kombination aus sowohl der „Warteschlangenabarbeitung“-Richtlinie als auch der „Nutzungshäufigkeit“-Richtlinie. Beispielsweise lässt der Speichermechanismus 406 das Hinzufügen einer Eigner-Speicherregion zu der vorab bestimmten Anzahl N an Eigner-Speicherregionen gemäß der „Warteschlangenabarbeitung“-Richtlinie zu. Anschließend können mittels des der „Nutzungshäufigkeit“-Richtlinie zugehörigen Grenzwerts weitere Eigner-Speicherregionen M auf Grundlage der Nutzungshäufigkeit der Eigner-Regionen hinzugefügt werden, bis eine vorab bestimmte Anzahl N+M an Eigner-Speicherregionen vorhanden ist. Falls jedoch der Speichermechanismus 406 so konfiguriert ist, dass er nur die vorab bestimmte Anzahl N an Eigner-Speicherregionen verwendet, dann kann, sobald gemäß der „Warteschlangenabarbeitung“-Richtlinie die vorab bestimmte Anzahl N an Eigner-Speicherregionen erreicht ist, der Speichermechanismus 406 den der „Nutzungshäufigkeit“-Richtlinie zugehörigen Grenzwert verwenden, um zu ermitteln, ob eine (neue) potentielle Eigner-Speicherregion einträglich genug ist, um als Ersatz für eine bereits in der vorab bestimmten Anzahl N an Eigner-Speicherregionen vorhandene Eigner-Speicherregion hinzugefügt zu werden.

[0043] Das heißt, um ein Ersetzen weniger häufig genutzter Eigner durch häufiger genutzte zu ermöglichen, gewährleistet der Speichermechanismus 406 ein Abtrennen weniger häufig genutzter Eigner-Speicherregionen, um Raum für häufiger genutzte bzw. vorteilhaftere Eigner-Speicherregionen zu schaffen. Dies ist besonders wertvoll, wenn die Anzahl an Eignern sich der vorab bestimmten Anzahl N an Eigner-Speicherregionen nähert oder diese bereits erreicht hat. Auch hier unterhält der Speichermechanismus 406 über einen Zähler, eine Datenstruktur oder dergleichen einen Nachverfolgungswert zur Nutzungshäufigkeit jeder Eigner-Speicherregion unabhängig davon, ob die Eigner-Speicherregion Teil der vorab bestimmten Anzahl N an Eigner-Speicherregionen ist oder nicht.

[0044] Auf Grundlage dieser Informationen kann entdeckt werden, dass eine Eigner-Speicherregion, die nicht Teil der vorab bestimmten Anzahl N an Eigner-Speicherregionen ist, einen Nutzungshäufigkeitswert aufweist, der höher ist als der einer Eigner-Speicherregion, die Teil der vorab bestimmten Anzahl N an Eigner-Speicherregionen ist. Falls eine häufiger genutzte Eigner-Speicherregion identifiziert wird, wird die am wenigsten genutzte Eigner-Region in der vorab bestimmten Anzahl N an Eigner-Speicherregionen entfernt und die aktuell nicht in der vorab bestimmten Anzahl N an Eigner-Speicherregionen vorhandene, häufiger genutzte Eigner-Region hinzugefügt. Es ist zu beachten, dass die Datenblöcke in der entfernten Eigner-Speicherregion möglicherweise in eine der Eigner-Speicherregionen in der vorab bestimmten Anzahl N an Eigner-Speicherregionen geschrieben werden müssen. Das heißt, es kann dazu kommen, dass der Speichermechanismus 406 Duplikatkopien der 8-KB-Datensegmente, die sich in der entfernten Eigner-Speicherregion befanden, in eine der aktuell in der vorab bestimmten Anzahl N an Eigner-Speicherregionen vorhandenen Eigner-Speicherregionen schreibt. Zwar kann dies das doppelt vorhandene 8-KB-Segment innerhalb der Referrer-Speicherregion 408 und den zugehörigen Eigner-Speicherregionen 410a-4104n erzeugen, jedoch verringert das Erzeugen des doppelt vorhandenen 8 KB-Segments die für den Speichermechanismus 406 anfallenden Verwaltungsdaten, indem nicht mehr als die vorab bestimmte Anzahl N an Eigner-Speicherregionen geladen wird. Unter Verwendung einer Kombination aus einer „Warteschlangenabarbeitung“-Richtlinie und einer „Nutzungshäufigkeit“-Richtlinie verringert somit der Speichermechanismus 406 Verwaltungsdaten im Speicher oder in der Verarbeitung weiter und ist nicht von der Größe der Nutzereingaben/ausgaben abhängig.

[0045] Bei der vorliegenden Erfindung kann es sich um ein System, ein Verfahren und/oder ein Computerprogrammprodukt handeln. Das Computerprogrammprodukt kann (ein) durch einen Computer lesbare(s) Speichermedium (oder -medien) beinhalten, auf dem/denen durch einen Computer lesbare Programmanweisungen gespeichert ist/sind, um einen Prozessor dazu zu veranlassen, Aspekte der vorliegenden Erfindung auszuführen.

[0046] Bei dem durch einen Computer lesbaren Speichermedium kann es sich um eine physische Einheit handeln, die Anweisungen zur Verwendung durch eine Einheit zur Ausführung von Anweisungen behalten und speichern kann. Bei dem durch einen Computer lesbaren Speichermedium kann es sich zum Beispiel um eine elektronische Speichereinheit, eine magnetische Speichereinheit, eine optische Speichereinheit, eine elektromagnetische Speichereinheit, eine Halbleiterspeichereinheit oder jede

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

[0047] Hierin beschriebene, durch einen Computer lesbare Programmanweisungen können von einem durch einen Computer lesbaren Speichermedium auf jeweilige Datenverarbeitungs-/Verarbeitungseinheiten oder über ein Netzwerk wie zum Beispiel das Internet, ein lokales Netzwerk, ein Weitverkehrsnetz und/oder ein drahtloses Netzwerk auf einen externen Computer oder eine externe Speichereinheit heruntergeladen werden. Das Netzwerk kann Kupferübertragungskabel, Lichtwellenübertragungsleiter, drahtlose Übertragung, Leitwegrechner, Firewalls, Vermittlungseinheiten, Gateway-Computer und/oder Edge-Server aufweisen. Eine Netzwerkkartentypkarte oder Netzwerkschnittstelle in jeder Datenverarbeitungs-/Verarbeitungseinheit empfängt durch einen Computer lesbare Programmanweisungen aus dem Netzwerk und leitet die durch einen Computer lesbaren Programmanweisungen zur Speicherung in einem durch einen Computer lesbaren Speichermedium innerhalb der entsprechenden Datenverarbeitungs-/Verarbeitungseinheit weiter.

[0048] Bei durch einen Computer lesbaren Programmanweisungen zum Ausführen von Arbeitsschritten der vorliegenden Erfindung kann es sich um Assembler-Anweisungen, ISA-Anweisungen (Instruction-Set-Architecture), Maschinenanweisungen, maschinenabhängige Anweisungen, Mikrocode, Firmware-Anweisungen, zustandssetzende Daten oder entweder Quellcode oder Objektcode handeln, die in einer beliebigen Kombination aus einer oder mehreren Programmiersprachen geschrieben werden, darunter objektorientierte Programmiersprachen wie Java, Smalltalk, C++ o.ä.

sowie herkömmliche prozedurale Programmiersprachen wie die Programmiersprache „C“ oder ähnliche Programmiersprachen. Die durch einen Computer lesbaren Programmanweisungen können vollständig auf dem Computer des Benutzers, teilweise auf dem Computer des Benutzers, als eigenständiges Software-Paket, teilweise auf dem Computer des Benutzers und teilweise auf einem fernen Computer oder vollständig auf dem fernen Computer oder Server ausgeführt werden. In letzterem Fall kann der entfernt angeordnete Computer mit dem Computer des Benutzers durch eine beliebige Art Netzwerk verbunden sein, darunter ein lokales Netzwerk (LAN) oder ein Weitverkehrsnetz (WAN), oder die Verbindung kann mit einem externen Computer hergestellt werden (zum Beispiel über das Internet unter Verwendung eines Internet-Diensteanbieters). In einigen Ausführungsformen können elektronische Schaltungen, darunter zum Beispiel programmierbare Logikschaltungen, vor Ort programmierbare Gatter-Anordnungen (FPGA, field programmable gate arrays) oder programmierbare Logikanordnungen (PLA, programmable logic arrays) die durch einen Computer lesbaren Programmanweisungen ausführen, indem sie Zustandsinformationen der durch einen Computer lesbaren Programmanweisungen nutzen, um die elektronischen Schaltungen zu personalisieren, um Aspekte der vorliegenden Erfindung durchzuführen.

[0049] Aspekte der vorliegenden Erfindung sind hierin unter Bezugnahme auf Ablaufpläne und/oder Blockschaltbilder bzw. Schaubilder von Verfahren, Vorrichtungen (Systemen) und Computerprogrammprodukten gemäß Ausführungsformen der Erfindung beschrieben. Es wird darauf hingewiesen, dass jeder Block der Ablaufpläne und/oder der Blockschaltbilder bzw. Schaubilder sowie Kombinationen von Blöcken in den Ablaufplänen und/oder den Blockschaltbildern bzw. Schaubildern mittels durch einen Computer lesbarer Programmanweisungen ausgeführt werden können.

[0050] Diese durch einen Computer lesbaren Programmanweisungen können einem Prozessor eines Universalcomputers, eines Spezialcomputers oder einer anderen programmierbaren Datenverarbeitungsvorrichtung bereitgestellt werden, um eine Maschine zu erzeugen, so dass die über den Prozessor des Computers bzw. der anderen programmierbaren Datenverarbeitungsvorrichtung ausgeführten Anweisungen ein Mittel zur Umsetzung der in dem Block bzw. den Blöcken der Ablaufpläne und/oder der Blockschaltbilder bzw. Schaubilder festgelegten Funktionen/Schritte erzeugen. Diese durch einen Computer lesbaren Programmanweisungen können auch auf einem durch einen Computer lesbaren Speichermedium gespeichert sein, das einen Computer, eine programmierbare Datenverarbeitungsvorrichtung und/oder andere Einheiten so steuern kann, dass sie auf eine bestimmte Art funktionieren,

so dass das durch einen Computer lesbare Speichermedium, auf dem Anweisungen gespeichert sind, ein Herstellungsprodukt aufweist, darunter Anweisungen, welche Aspekte der/des in dem Block bzw. den Blöcken des Ablaufplans und/oder der Blockschaltbilder bzw. Schaubilder angegebenen Funktion/Schritts umsetzen.

[0051] Die durch einen Computer lesbaren Programmanweisungen können auch auf einen Computer, eine andere programmierbare Datenverarbeitungsvorrichtung oder eine andere Einheit geladen werden, um das Ausführen einer Reihe von Prozessschritten auf dem Computer bzw. der anderen programmierbaren Vorrichtung oder anderen Einheit zu verursachen, um einen auf einem Computer ausgeführten Prozess zu erzeugen, so dass die auf dem Computer, einer anderen programmierbaren Vorrichtung oder einer anderen Einheit ausgeführten Anweisungen die in dem Block bzw. den Blöcken der Ablaufpläne und/oder der Blockschaltbilder bzw. Schaubilder festgelegten Funktionen/Schritte umsetzen.

[0052] Fig. 5 zeigt ein Flussdiagramm der Arbeitsschritte, die von einem Speichermechanismus durchgeführt werden, der gemäß einer veranschaulichenden Ausführungsform die Leistung verteilter, ortsbezogener Deduplikation mit minimaler Auswirkung auf die Deduplikationseffizienz verbessert. Der Vorgang beginnt damit, dass der Speichermechanismus eine Anfrage zum Schreiben einer Datendatei in eine Speicherregion innerhalb einer Menge aus Speicherregionen empfängt (Schritt 502). Für jedes Datensegment der Datendatei erzeugt der Speichermechanismus einen Hashwert für das Datensegment (Schritt 504) und vergleicht den Hashwert mit Hashwerten für andere in der Menge aus Speicherregionen referenzierte gespeicherte Datensegmente (Schritt 506). Falls in Schritt 506 der Vergleich anzeigt, dass das Datensegment bereits in den Referrer-Speicherregionen oder einer der der Referrer-Speicherregion zugehörigen Eigner-Speicherregionen vorhanden ist, speichert der Speichermechanismus einen Zeiger auf dieses Datensegment (Schritt 508), und der Vorgang endet. Falls in Schritt 506 der Vergleich anzeigt, dass das Datensegment nicht bereits in den beschriebenen Speicherregionen vorhanden ist, d.h. in den Referrer-Speicherregionen oder der einen oder den mehreren zugehörigen Eigner-Speicherregionen, speichert der Speichermechanismus das Datensegment in der Referrer-Speicherregion (Schritt 510), und der Vorgang endet.

[0053] Fig. 6 zeigt ein Flussdiagramm der Arbeitsschritte, die gemäß einer veranschaulichenden Ausführungsform von einem Speichermechanismus durchgeführt werden, wenn Deduplikation unter einer Teilmenge aus Speicherregionen, d.h. den Referrer-Speicherregionen und der einen oder den

mehreren zugehörigen Eigner-Speicherregionen, umgesetzt werden soll. Vor dem Speichern des Datensegments einer Datendatei in einer Referrer-Region, in welche die Daten durch das Hostsystem geschrieben werden, ermittelt der Speichermechanismus, ob die Daten in einer anderen Speicherregion in der Menge aus Speicherregionen vorhanden sind (Schritt 602). Falls in Schritt 602 die Daten nicht in einer anderen Speicherregion in der Menge aus Speicherregionen vorhanden sind, speichert der Speichermechanismus die Daten in der Referrer-Speicherregion (Schritt 604), und der Vorgang endet.

[0054] Falls in Schritt 602 die Daten in einer anderen Speicherregion in der Menge aus Speicherregionen vorhanden sind, ermittelt die Speichersteuerung, ob eine vorab bestimmte Anzahl N an Eigner-Speicherregionen erreicht wurde (Schritt 606). Falls in Schritt 606 die vorab bestimmte Anzahl N an Eigner-Regionen erreicht wurde, speichert der Speichermechanismus das Datensegment in der Referrer-Speicherregion in der vorab bestimmten Anzahl N an Eigner-Speicherregionen (Schritt 608), und der Vorgang endet. Falls in Schritt 606 die vorab bestimmte Anzahl N an Eigner-Speicherregionen nicht erreicht wurde, speichert der Speichermechanismus einen Verweis auf das Datensegment in der Referrer-Speicherregion (Schritt 610). Der Speichermechanismus kann dann die Speicherregion der vorab bestimmten Anzahl N an Speicherregionen hinzufügen, je nachdem, gemäß welcher Richtlinie der Speichermechanismus aktuell arbeitet (Schritt 612), und der Prozess endet.

[0055] Die Ablaufpläne und die Blockschaltbilder bzw. Schaubilder in den Figuren veranschaulichen die Architektur, die Funktionalität und den Betrieb möglicher Ausführungen von Systemen, Verfahren und Computerprogrammprodukten gemäß verschiedenen Ausführungsformen der vorliegenden Erfindung. In diesem Zusammenhang kann jeder Block in den Ablaufplänen oder Blockschaltbildern bzw. Schaubildern ein Modul, ein Segment oder einen Teil von Anweisungen darstellen, die eine oder mehrere ausführbare Anweisungen zur Ausführung der bestimmten logischen Funktion(en) aufweisen. In einigen alternativen Ausführungen können die in dem Block angegebenen Funktionen in einer anderen Reihenfolge als in den Figuren gezeigt stattfinden. Zwei nacheinander gezeigte Blöcke können zum Beispiel in Wirklichkeit im Wesentlichen gleichzeitig ausgeführt werden, oder die Blöcke können manchmal je nach entsprechender Funktionalität in umgekehrter Reihenfolge ausgeführt werden. Es ist ferner anzumerken, dass jeder Block der Blockschaltbilder bzw. Schaubilder und/oder der Ablaufpläne sowie Kombinationen aus Blöcken in den Blockschaltbildern bzw. Schaubildern und/oder den Ablaufplänen durch spezielle auf Hardware beruhende Systeme umgesetzt werden können, welche

die festgelegten Funktionen oder Schritte durchführen, oder Kombinationen aus Spezial-Hardware und Computeranweisungen ausführen.

[0056] Somit stellen die veranschaulichenden Ausführungsformen Mechanismen zum Verbessern der Leistung verteilter, ortsbezogener Deduplikation mit minimaler Auswirkung auf die Deduplikationseffizienz bereit. Durch intelligentes Erstellen von Deduplikationsverknüpfungen zwischen verschiedenen Referrer- und Eigner-Speicherregionen werden Speicheranforderungen zum Lesen aller referenzierter Daten mit minimaler Auswirkung auf das Deduplikationsverhältnis minimiert, die Menge an Metadatenregionen für Umlagerung mit minimaler Auswirkung auf das Deduplikationsverhältnis minimiert, die Menge an Datenaustausch zwischen dem Speichermechanismus und den Eigner-Speicherregionen mit minimaler Auswirkung auf das Deduplikationsverhältnis minimiert und/oder differenzierte Steuerung der Verteilung von Deduplikationsverknüpfungen über die Metadatenregionen des Systems bereitgestellt.

[0057] Wie vorstehend erwähnt, ist zu beachten, dass die veranschaulichenden Ausführungsformen in Form einer reinen Hardware-Ausführungsform, einer reinen Software-Ausführungsform oder einer sowohl Hardware- als auch Software-Elemente enthaltenden Ausführungsform vorliegen können. In einer beispielhaften Ausführungsform werden die Mechanismen der veranschaulichenden Ausführungsformen in Software oder Programmcode implementiert, darunter, ohne jedoch hierauf eingeschränkt zu sein, Firmware, systemeigene Software, Mikrocode usw.

[0058] Ein zum Speichern und/oder Ausführen von Programmcode geeignetes Datenverarbeitungssystem beinhaltet mindestens einen Prozessor, der durch einen Datenübertragungsbus wie beispielsweise einen Systembus direkt oder indirekt mit Arbeitsspeicherelementen verbunden ist. Zu den Arbeitsspeicherelementen können während der eigentlichen Ausführung des Programmcodes eingesetzter lokaler Arbeitsspeicher, Großraumspeicher und Cachespeicher zählen, die eine kurzzeitige Speicherung von zumindest einigem Programmcode gewährleisten, um die Zahl von Abrufen von Code aus dem Großraumspeicher während der Ausführung zu verringern. Bei dem Arbeitsspeicher kann es sich um verschiedene Arten handeln, darunter, ohne jedoch hierauf eingeschränkt zu sein, ROM, PROM, EPROM, EEPROM, DRAM, SRAM, Flashspeicher, Solid-State-Speicher und dergleichen.

[0059] Eingabe/Ausgabe- bzw. E/A-Einheiten (darunter, ohne jedoch hierauf eingeschränkt zu sein, Tastaturen, Anzeigen, Zeigeeinheiten usw.) können entweder direkt oder durch dazwischenlie-

gende drahtgebundene oder drahtlose E/A-Schnittstellen und/oder Steuerungen oder dergleichen mit dem System verbunden sein. E/A-Einheiten können neben herkömmlichen Tastaturen, Anzeigen, Zeigeeinheiten und dergleichen in vielen verschiedenen Formen vorliegen, beispielsweise Datenübertragungseinheiten, die durch drahtgebundene oder drahtlose Verbindungen verbunden sind, darunter, ohne jedoch hierauf eingeschränkt zu sein, Smartphones, Tablet-Computer, Einheiten mit berührungsempfindlichen Bildschirmen, Spracherkennungseinheiten und dergleichen. Jede bekannte oder künftig entwickelte E/A-Einheit soll vom Umfang der veranschaulichenden Ausführungsformen umfasst sein.

[0060] Mit dem System können auch Netzwerkadapter verbunden sein, um eine Verbindung des Datenverarbeitungssystems mit anderen Datenverarbeitungssystemen oder entfernt angeordneten Druckern oder Speichereinheiten durch dazwischenliegende private oder öffentliche Netzwerke zu ermöglichen. Modems, Kabelmodems und Ethernetkarten sind nur einige der aktuell erhältlichen Arten von Netzwerkadaptern für drahtgebundene Nachrichtenübermittlung. Es können auch auf drahtloser Datenübertragung beruhende Netzwerkadapter genutzt werden, darunter, ohne jedoch hierauf eingeschränkt zu sein, Adapter für drahtlose Kommunikation gemäß 802.11 a/b/g/n, drahtlose Bluetooth-Adapter und dergleichen. Alle bekannten oder künftig entwickelten Netzwerkadapter sollen vom Grundgedanken und Umfang der vorliegenden Erfindung umfasst sein.

Patentansprüche

1. Verfahren in einem Datenverarbeitungssystem (200; 400) für verteilte, ortsbezogene Datenspeicherung, wobei das Verfahren Folgendes umfasst:

durch einen Speichermechanismus (406) erfolgreiches Empfangen (502) einer Anfrage von einem Hostsystem (102; 402) zum Schreiben einer Datendatei in eine Referrer-Speicherregion (408) in einer Menge aus Speicherregionen (404a, 404b, 404c, 404d, 404n, 404max) und

für jedes Datensegment der Datendatei:

in Reaktion darauf, dass ein Vergleich (506) eines Hashwerts für das Datensegment mit anderen Hashwerten für andere in der Referrer-Speicherregion referenzierte gespeicherte Datensegmente anzeigt, dass das Datensegment in der Referrer-Speicherregion nicht vorhanden ist, durch den Speichermechanismus erfolgreiches Ermitteln (602), ob die Daten in einer anderen Speicherregion in der Menge aus Speicherregionen vorhanden sind,

in Reaktion darauf, dass das Datensegment in einer anderen Speicherregion in der Menge aus Speicherregionen vorhanden ist, durch den Speichermechanismus erfolgreiches Ermitteln, ob es sich bei der

Speicherregion um eine aus einer vorab bestimmten Anzahl N an der Referrer-Speicherregion zugehörigen Eigner-Speicherregionen (410a, 410b, 410c, 410n) handelt,

in Reaktion darauf, dass es sich bei der Speicherregion nicht um eine aus der vorab bestimmten Anzahl N an der Referrer-Speicherregion zugehörigen Eigner-Speicherregionen handelt, durch den Speichermechanismus erfolgreiches Ermitteln, ob die Anzahl an Eigner-Speicherregionen in der vorab bestimmten Anzahl N an Eigner-Speicherregionen erreicht wurde (606), und in Reaktion darauf, dass die vorab bestimmte Anzahl N an Eigner-Speicherregionen nicht erreicht wurde, durch den Speichermechanismus erfolgreiches Speichern (610) eines Verweises auf das Datensegment in der Referrer-Speicherregion.

2. Verfahren nach Anspruch 1, ferner umfassend:

in Reaktion darauf, dass die vorab bestimmte Anzahl N an Eigner-Speicherregionen erreicht wurde, durch den Speichermechanismus erfolgreiches Speichern (608) des Datensegments in der Referrer-Speicherregion.

3. Verfahren nach Anspruch 1, ferner umfassend:

in Reaktion darauf, dass die vorab bestimmte Anzahl N an Eigner-Speicherregionen nicht erreicht wurde, durch den Speichermechanismus erfolgreiches Ermitteln, ob die Speicherregion, in der das Datenelement gespeichert wird, einen Nutzungshäufigkeitsgrenzwert erreicht hat, und in Reaktion darauf, dass die Speicherregion, in der das Datensegment gespeichert wurde, den Nutzungshäufigkeitsgrenzwert erreicht hat, durch den Speichermechanismus erfolgreiches Hinzufügen (612) der Speicherregion zu der vorab bestimmten Anzahl N an Eigner-Speicherregionen.

4. Verfahren nach Anspruch 1, ferner umfassend:

in Reaktion darauf, dass die vorab bestimmte Anzahl N an Eigner-Speicherregionen nicht erreicht wurde, durch den Speichermechanismus erfolgreiches Hinzufügen (612) der Speicherregion zu der vorab bestimmten Anzahl N an Eigner-Speicherregionen.

5. Verfahren nach Anspruch 1, wobei die Eigner-Speicherregionen innerhalb der vorab bestimmten Anzahl N an Eigner-Speicherregionen auf Grundlage entweder einer „Warteschlangenabarbeitung“-Richtlinie oder einer „Nutzungshäufigkeit“-Richtlinie bestimmt werden.

6. Verfahren nach Anspruch 5, wobei gemäß der „Warteschlangenabarbeitung“-Richtlinie die Speicherregion der vorab bestimmten Anzahl N an Eig-

ner-Speicherregionen hinzugefügt wird, wenn Daten in der Speicherregion gespeichert werden.

7. Verfahren nach Anspruch 5, wobei gemäß der „Nutzungshäufigkeit“-Richtlinie die Speicherregion der vorab bestimmten Anzahl N an Eigner-Speicherregionen hinzugefügt wird, wenn die Nutzungshäufigkeit der Speicherregionen einen vorab bestimmten Grenzwert erreichen, wobei das Erreichen des vorab bestimmten Grenzwerts vorliegt, wenn eine Anzahl an Lese- und/oder Schreibvorgängen in die Speicherregion den Grenzwert erreicht oder überschreitet.

8. Verfahren nach Anspruch 7, wobei gemäß der „Nutzungshäufigkeit“-Richtlinie eine Eigner-Speicherregion aus der vorab bestimmten Anzahl N an Eigner-Speicherregionen entfernt wird, wenn eine Nutzungshäufigkeit einer anderen Speicherregion die Nutzungshäufigkeit einer am wenigsten genutzten Eigner-Speicherregion überschreitet, die sich bereits in der vorab bestimmten Anzahl N an Eigner-Speicherregionen befindet, wenn die vorab bestimmte Anzahl N an Eigner-Speicherregionen voll ist.

9. Computerprogrammprodukt, welches ein computerlesbares Speichermedium mit einem auf diesem gespeicherten computerlesbaren Programm aufweist, wobei das computerlesbare Programm bei Ausführung auf einer Datenverarbeitungseinheit (200; 400) die Datenverarbeitungseinheit veranlasst:

von einem Hostsystem (102; 402) eine Anfrage zum Schreiben einer Datendatei in eine Referrer-Speicherregion (408) in einer Menge aus Speicherregionen (404a, 404b, 404c, 404d, 404n, 404max) zu empfangen (502) und

für jedes Datensegment der Datendatei:

in Reaktion darauf, dass ein Vergleich (506) eines Hashwerts für das Datensegment mit anderen Hashwerten für andere in der Referrer-Speicherregion referenzierte gespeicherte Datensegmente anzeigt, dass das Datensegment in der Referrer-Speicherregion nicht vorhanden ist, zu ermitteln (602), ob die Daten in einer anderen Speicherregion in der Menge aus Speicherregionen vorhanden sind, in Reaktion darauf, dass das Datensegment in einer anderen Speicherregion in der Menge aus Speicherregionen vorhanden ist, zu ermitteln, ob es sich bei der Speicherregion um eine aus einer vorab bestimmten Anzahl N an der Referrer-Speicherregion zugehörigen Eigner-Speicherregionen (410a, 410b, 410c, 410n) handelt, in Reaktion darauf, dass es sich bei der Speicherregion nicht um eine aus der vorab bestimmten Anzahl N an der Referrer-Speicherregion zugehörigen Eigner-Speicherregionen handelt, zu ermitteln, ob die Anzahl an Eigner-Speicherregionen in der vorab bestimmten Anzahl N an Eigner-Speicherre-

gionen erreicht wurde (606), und in Reaktion darauf, dass die vorab bestimmte Anzahl N an Eigner-Speicherregionen nicht erreicht wurde, einen Verweis auf das Datensegment in der Referrer-Speicherregion zu speichern (610).

10. Computerprogrammprodukt nach Anspruch 9, wobei das computerlesbare Programm ferner die Datenverarbeitungseinheit veranlasst: in Reaktion darauf, dass die vorab bestimmte Anzahl N an Eigner-Speicherregionen erreicht wurde, das Datensegment in der Referrer-Speicherregion zu speichern (608).

11. Computerprogrammprodukt nach Anspruch 9, wobei das computerlesbare Programm ferner die Datenverarbeitungseinheit veranlasst: in Reaktion darauf, dass die vorab bestimmte Anzahl N an Eigner-Speicherregionen nicht erreicht wurde, zu ermitteln, ob die Speicherregion, in der das Datenelement gespeichert wird, einen Nutzungshäufigkeitsgrenzwert erreicht hat, und in Reaktion darauf, dass die Speicherregion, in der das Datensegment gespeichert wurde, den Nutzungshäufigkeitsgrenzwert erreicht hat, die Speicherregion zu der vorab bestimmten Anzahl N an Eigner-Speicherregionen hinzuzufügen (612).

12. Computerprogrammprodukt nach Anspruch 9, wobei das computerlesbare Programm ferner die Datenverarbeitungseinheit veranlasst: in Reaktion darauf, dass die vorab bestimmte Anzahl N an Eigner-Speicherregionen nicht erreicht wurde, die Speicherregion zu der vorab bestimmten Anzahl N an Eigner-Speicherregionen hinzuzufügen (612).

13. Computerprogrammprodukt nach Anspruch 9, wobei die Eigner-Speicherregionen innerhalb der vorab bestimmten Anzahl N an Eigner-Speicherregionen auf Grundlage entweder einer „Warteschlangenabarbeitung“-Richtlinie oder einer „Nutzungshäufigkeit“-Richtlinie bestimmt werden.

14. Computerprogrammprodukt nach Anspruch 13, wobei: gemäß der „Warteschlangenabarbeitung“-Richtlinie die Speicherregion der vorab bestimmten Anzahl N an Eigner-Speicherregionen hinzugefügt wird, wenn Daten in der Speicherregion gespeichert werden, gemäß der „Nutzungshäufigkeit“-Richtlinie die Speicherregion der vorab bestimmten Anzahl N an Eigner-Speicherregionen hinzugefügt wird, wenn die Nutzungshäufigkeit der Speicherregionen einen vorab bestimmten Grenzwert erreichen, wobei das Erreichen des vorab bestimmten Grenzwerts vorliegt, wenn eine Anzahl an Lese- und/oder Schreibvorgängen in die Speicherregion den Grenzwert erreicht oder überschreitet, oder gemäß der „Nutzungshäufigkeit“-Richtlinie eine Eig-

ner-Speicherregion aus der vorab bestimmten Anzahl N an Eigner-Speicherregionen entfernt wird, wenn eine Nutzungshäufigkeit einer anderen Speicherregion die Nutzungshäufigkeit einer am wenigsten genutzten Eigner-Speicherregion überschreitet, die sich bereits in der vorab bestimmten Anzahl N an Eigner-Speicherregionen befindet, wenn die vorab bestimmte Anzahl N an Eigner-Speicherregionen voll ist.

15. Vorrichtung (200; 400), die Folgendes aufweist: einen Prozessor und einen mit dem Prozessor verbundenen Arbeitsspeicher, wobei der Arbeitsspeicher Anweisungen aufweist, die bei Ausführung durch den Prozessor den Prozessor veranlassen: von einem Hostsystem (102; 402) eine Anfrage zum Schreiben einer Datendatei in eine Referrer-Speicherregion (408) in einer Menge aus Speicherregionen (404a, 404b, 404c, 404d, 404n, 404max) zu empfangen (502) und für jedes Datensegment der Datendatei: in Reaktion darauf, dass ein Vergleich (506) eines Hashwerts für das Datensegment mit anderen Hashwerten für andere in der Referrer-Speicherregion referenzierte gespeicherte Datensegmente anzeigt, dass das Datensegment in der Referrer-Speicherregion nicht vorhanden ist, zu ermitteln (602), ob die Daten in einer anderen Speicherregion in der Menge aus Speicherregionen vorhanden sind, in Reaktion darauf, dass das Datensegment in einer anderen Speicherregion in der Menge aus Speicherregionen vorhanden ist, zu ermitteln, ob es sich bei der Speicherregion um eine aus einer vorab bestimmten Anzahl N an der Referrer-Speicherregion zugehörigen Eigner-Speicherregionen (410a, 410b, 410c, 410n) handelt, in Reaktion darauf, dass es sich bei der Speicherregion nicht um eine aus der vorab bestimmten Anzahl N an der Referrer-Speicherregion zugehörigen Eigner-Speicherregionen handelt, zu ermitteln, ob die Anzahl an Eigner-Speicherregionen in der vorab bestimmten Anzahl N an Eigner-Speicherregionen erreicht wurde (606), und in Reaktion darauf, dass die vorab bestimmte Anzahl N an Eigner-Speicherregionen nicht erreicht wurde, einen Verweis auf das Datensegment in der Referrer-Speicherregion zu speichern (610).

16. Vorrichtung nach Anspruch 15, wobei die Anweisungen den Prozessor ferner veranlassen: in Reaktion darauf, dass die vorab bestimmte Anzahl N an Eigner-Speicherregionen erreicht wurde, das Datensegment in der Referrer-Speicherregion zu speichern (608).

17. Vorrichtung nach Anspruch 15, wobei die Anweisungen den Prozessor ferner veranlassen: in Reaktion darauf, dass die vorab bestimmte

Anzahl N an Eigner-Speicherregionen nicht erreicht wurde, zu ermitteln, ob die Speicherregion, in der das Datenelement gespeichert wird, einen Nutzungshäufigkeitsgrenzwert erreicht hat, und in Reaktion darauf, dass die Speicherregion, in der das Datensegment gespeichert wurde, den Nutzungshäufigkeitsgrenzwert erreicht hat, die Speicherregion zu der vorab bestimmten Anzahl N an Eigner-Speicherregionen hinzuzufügen (612).

18. Vorrichtung nach Anspruch 15, wobei die Anweisungen den Prozessor ferner veranlassen: in Reaktion darauf, dass die vorab bestimmte Anzahl N an Eigner-Speicherregionen nicht erreicht wurde, die Speicherregion zu der vorab bestimmten Anzahl N an Eigner-Speicherregionen hinzuzufügen (612).

19. Vorrichtung nach Anspruch 15, wobei die Eigner-Speicherregionen innerhalb der vorab bestimmten Anzahl N an Eigner-Speicherregionen auf Grundlage entweder einer „Warteschlangenabarbeitung“-Richtlinie oder einer „Nutzungshäufigkeit“-Richtlinie bestimmt werden.

20. Vorrichtung nach Anspruch 19, wobei: gemäß der „Warteschlangenabarbeitung“-Richtlinie die Speicherregion der vorab bestimmten Anzahl N an Eigner-Speicherregionen hinzugefügt wird, wenn Daten in der Speicherregion gespeichert werden, gemäß der „Nutzungshäufigkeit“-Richtlinie die Speicherregion der vorab bestimmten Anzahl N an Eigner-Speicherregionen hinzugefügt wird, wenn die Nutzungshäufigkeit der Speicherregionen einen vorab bestimmten Grenzwert erreichen, wobei das Erreichen des vorab bestimmten Grenzwerts vorliegt, wenn eine Anzahl an Lese- und/oder Schreibvorgängen in die Speicherregion den Grenzwert erreicht oder überschreitet, oder gemäß der „Nutzungshäufigkeit“-Richtlinie eine Eigner-Speicherregion aus der vorab bestimmten Anzahl N an Eigner-Speicherregionen entfernt wird, wenn eine Nutzungshäufigkeit einer anderen Speicherregion die Nutzungshäufigkeit einer am wenigsten genutzten Eigner-Speicherregion überschreitet, die sich bereits in der vorab bestimmten Anzahl N an Eigner-Speicherregionen befindet, wenn die vorab bestimmte Anzahl N an Eigner-Speicherregionen voll ist.

Es folgen 5 Seiten Zeichnungen

Anhängende Zeichnungen

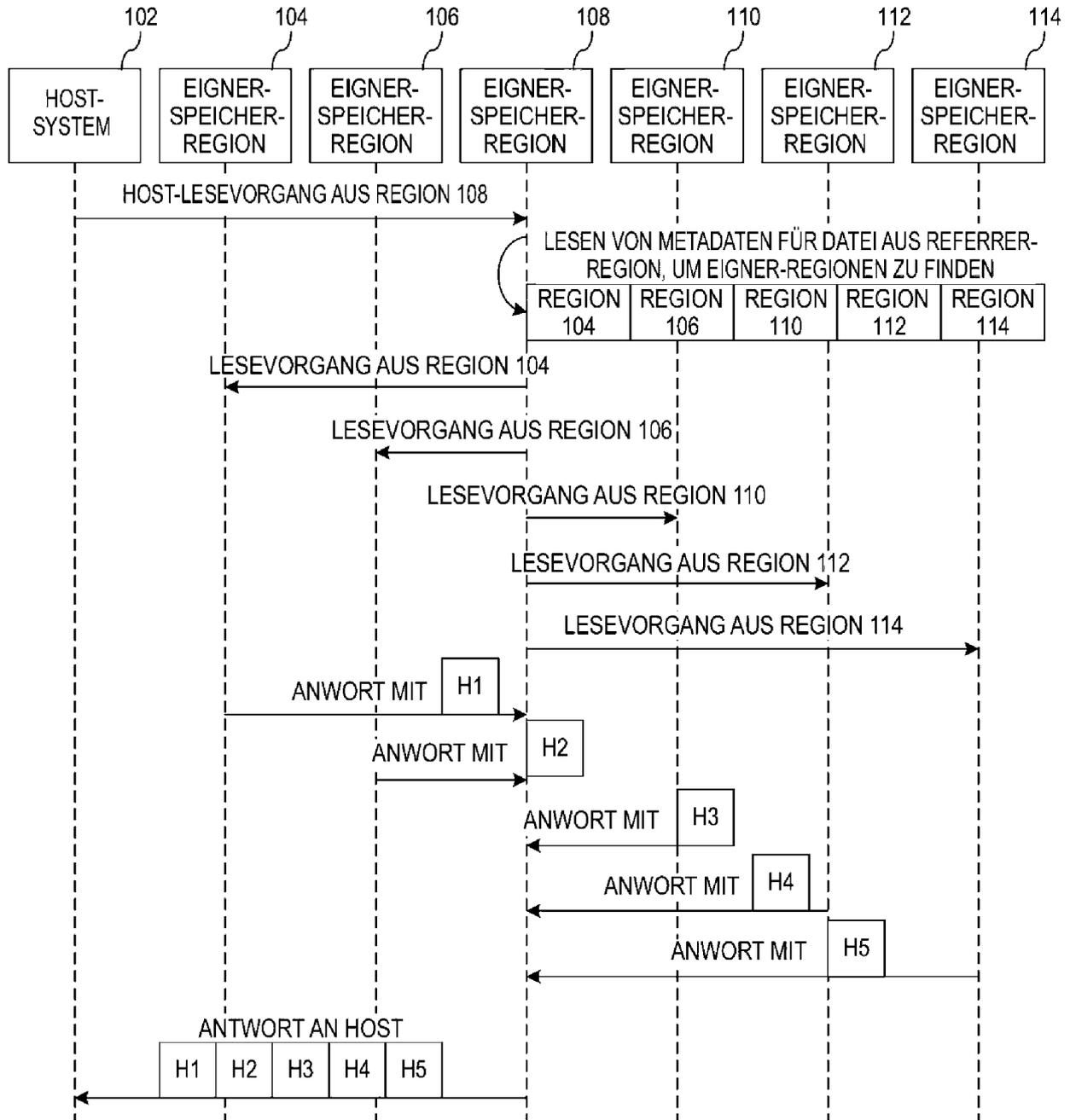


FIG. 1

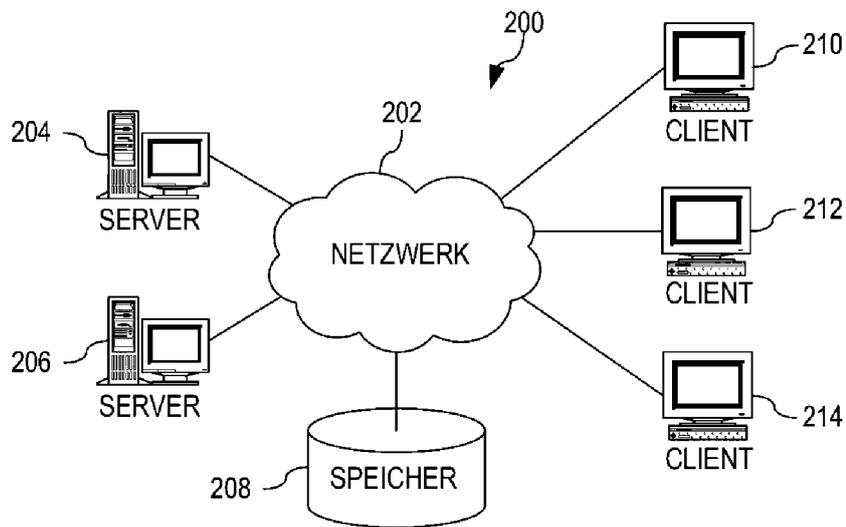


FIG. 2

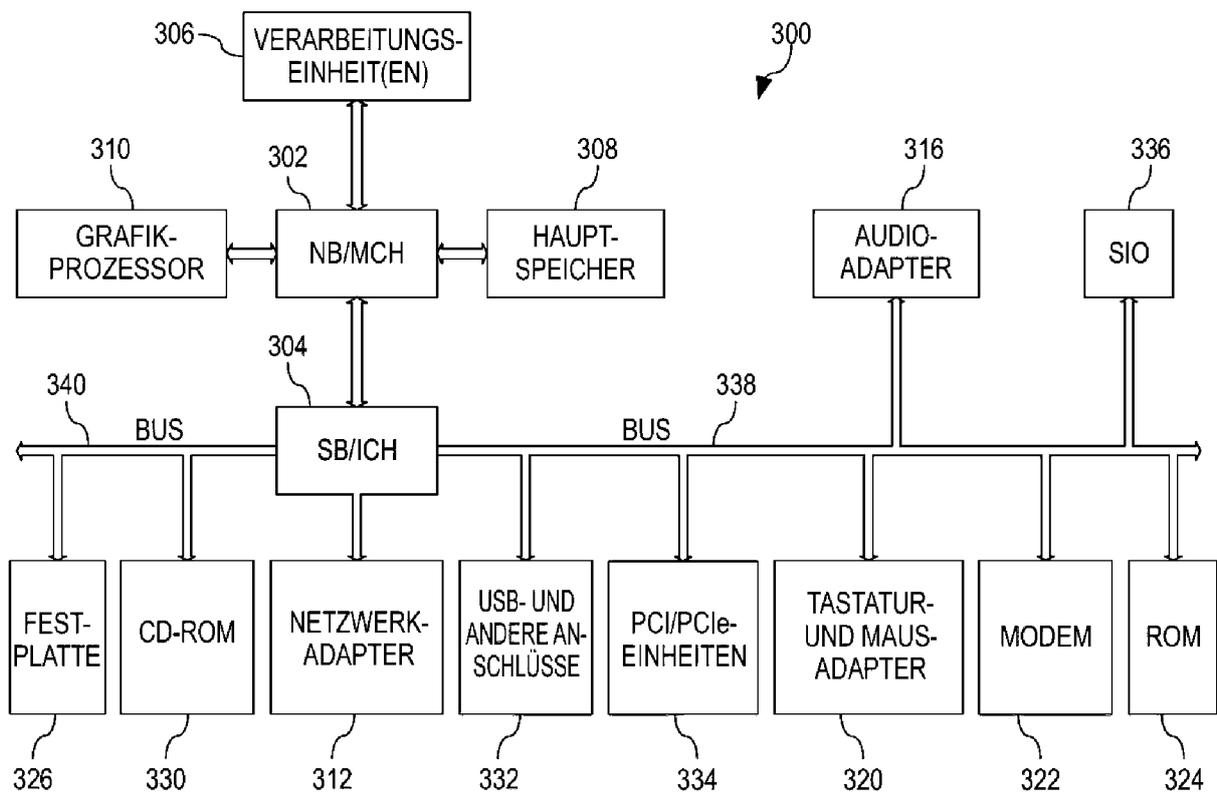


FIG. 3

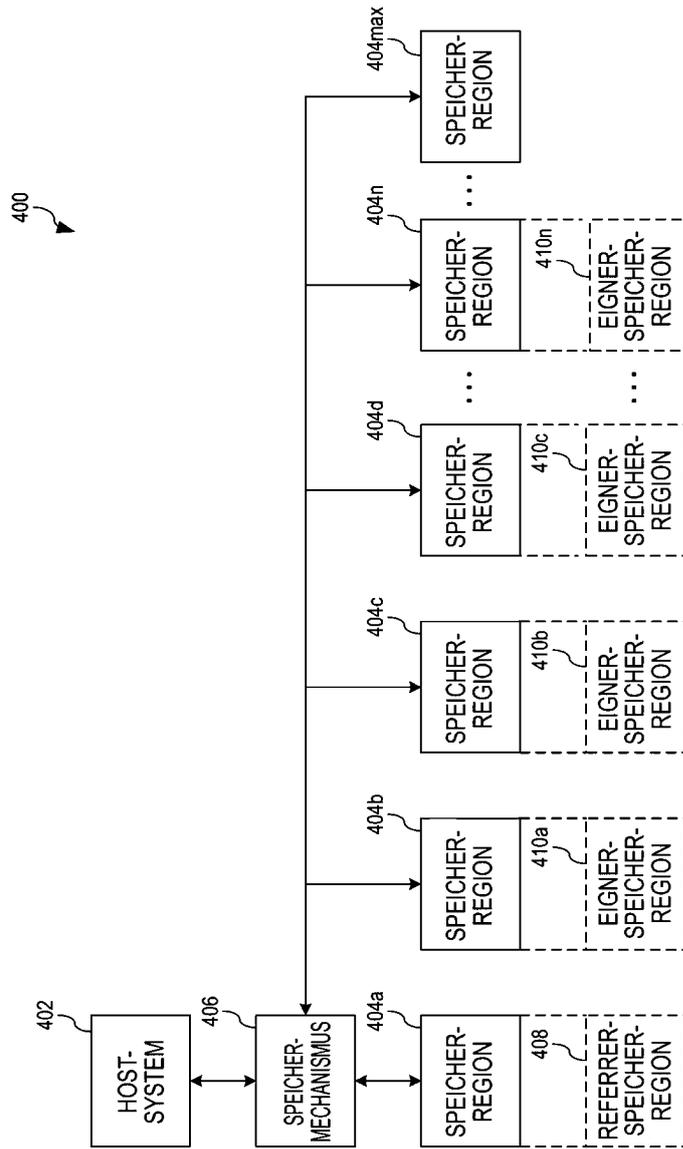


FIG. 4

