(19) Deutsches Patent- und Markenamt

(10) **DE 11 2018 004 178 B4** 2024.03.07

(12) # Patentschrift

(21) Deutsches Aktenzeichen: **11 2018 004 178.6**
(86) PCT-Aktenzeichen: **PCT/US2018/000337**
(87) PCT-Veröffentlichungs-Nr.: **WO 2019/036045**
(86) PCT-Anmeldetag: **17.08.2018**
(87) PCT-Veröffentlichungstag: **21.02.2019**
(43) Veröffentlichungstag der PCT Anmeldung in deutscher Übersetzung: **14.05.2020**
(45) Veröffentlichungstag der Patenterteilung: **07.03.2024**

(51) Int Cl.: ***G06F 16/13*** (2019.01)
G06F 16/14 (2019.01)
G06F 16/90 (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).

(30) Unionspriorität:

62/546,272	**16.08.2017**	**US**
15/999,199	**16.08.2018**	**US**

(73) Patentinhaber:
Hewlett Packard Enterprise Development LP, Spring, TX, US

(74) Vertreter:
HL Kempner Patentanwälte, Solicitors (England & Wales), Irish Patent Agents Partnerschaft mbB, 80538 München, DE

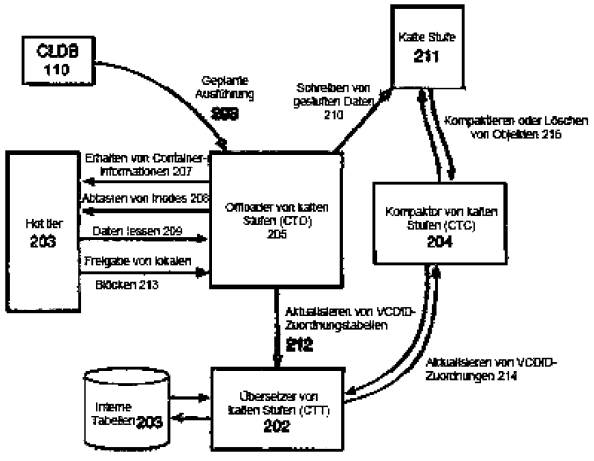
(72) Erfinder:
Saradhi, Uppaluri Vijaya, San Jose, CA, US; Pande, Arvind Arun, San Jose, CA, US; Rastogi, Kanishk, San Jose, CA, US; Reddy D, Giri Prasad, San Jose, CA, US; Bhupale, Nikhil, San Jose, CA, US; Boddu, Rajesh, San Jose, CA, US; Sanapala, Chandra Guru Kiran Babu, San Jose, CA, US; Jonnala, Premkumar, San Jose, CA, US; Sangwan, Ashish, San Jose, CA, US

(56) Ermittelter Stand der Technik:

US	**2014 / 0 379 715**	**A1**
US	**2016 / 0 041 907**	**A1**

(54) Bezeichnung: **MEHRSTUFIGE SPEICHERUNG IN EINEM VERTEILTEN DATEISYSTEM**

(57) Hauptanspruch: Verfahren, umfassend:
Empfangen, an einem Dateiserver (303), von einem Benutzergerät (301), einer Anforderung nach Daten, die durch einen virtuellen Cluster-Deskriptor dargestellt werden, wobei der Dateiserver (303) eine gestuftes Datenspeichersystem (100) umfasst;
Abfragen einer Kennungszuordnung unter Verwendung einer Kennung des virtuellen Cluster-Deskriptors;
als Reaktion darauf, dass die Kennungszuordnung angibt, dass die angeforderten Daten an einem vom Dateiserver (303) entfernten Ort gespeichert sind:
Senden einer Benachrichtigung zu dem Benutzergerät (301), um das Benutzergerät (301) dazu zu veranlassen die angeforderten Daten zu einem späteren Zeitpunkt von dem Dateiserver (303) anzufordern;
Zugreifen auf eine Übersetzungstabelle für kalte Stufen (203), in der eine Zuordnung zwischen einer Kennung eines jeden von mehreren virtuellen Cluster-Deskriptoren und einem Speicherort von Daten gespeichert ist, die dem jeweiligen virtuellen Cluster-Deskriptor zugeordnet sind, wobei das gestufte Datenspeichersystem (100) einen Speicher (205) für kalte Stufen umfasst, die zum Auslagern ausgewählt sind;
Abfragen der Übersetzungstabelle (203) für kalte Stufen unter Verwendung der Kennung des virtuellen Cluster-Deskriptors, der den angeforderten Daten zugeordnet ist, um einen Speicherort der angeforderten Daten in dem Speicher (205) für kalte Stufen zu identifizieren; und
Laden der angeforderten Daten auf den Dateiserver von dem identifizierten Speicherort.



Beschreibung

TECHNISCHES GEBIET

[0001] Verschiedene der offenbarten Ausführungsformen betreffen ein verteiltes Dateisystem und insbesondere eine mehrstufige Speicherung in einem verteilten Dateisystem.

HINTERGRUND

[0002] Unternehmen suchen nach Lösungen, die die widersprüchlichen Anforderungen an kostengünstige Speicher erfüllen, die sich häufig an Standorten außerhalb des Unternehmens befinden und gleichzeitig einen Hochgeschwindigkeits-Datenzugriff gewährleisten. Sie möchten auch eine praktisch unbegrenzte Speicherkapazität haben. Bei den derzeitigen Ansätzen muss ein Kunde häufig Produkte von Drittanbietern wie Cloud-Gateways kaufen, die ineffizient und teuer sind und Verwaltungs- und Anwendungskomplexität verursachen.

[0003] Es gibt einige zusätzliche Überlegungen, die in modernen Big-Data-Systemen auftreten, wenn versucht wird, kalte Daten auf eine kalte Speicherstufe zu übertragen, wobei „kalte“ oder „eingefrorene“ Daten solche Daten sind, auf die selten zugegriffen wird. Ein besonderer Aspekt vieler kostengünstiger Objektspeicher wie Amazon S3 oder Azure Object Store ist, dass die Objekte im Objektspeicher vorzugsweise relativ groß sein sollen (10 MB oder mehr). Es ist möglich, viel kleinere Objekte zu speichern, aber aus Gründen der Speichereffizienz, der Leistung und der Kosten werden Lösungen bevorzugt, bei denen größere Objekte verwendet werden.

[0004] In einem modernen Big-Data-System kann es beispielsweise eine sehr große Anzahl von Dateien geben. Einige dieser Systeme verfügen beispielsweise über mehr als eine Billion Dateien mit einer Dateierzeugungsrate von mehr als 2 Milliarden pro Tag, wobei die Erwartung besteht, dass diese Anzahl weiter zunehmen wird. In Systemen mit einer so großen Anzahl von Dateien sind die durchschnittliche und die mittlere Dateigröße notwendigerweise viel kleiner als die gewünschte Dateneinheit, die in den kalten, gestuften Speicher geschrieben wird. Bei einem System mit 1 PB Speicherplatz und einer Billion Dateien beträgt die durchschnittliche Dateigröße 1018/1012 = 1 MB und liegt damit deutlich unter der gewünschten Objektgröße. Darüber hinaus sind viele Systeme mit einer großen Anzahl von Dateien insgesamt erheblich kleiner als ein Petabyte und weisen durchschnittliche Dateigrößen von etwa 100 kB auf. Der S3 von Amazon verfügte erst 2014 über insgesamt zwei Billionen Objekte bei allen Nutzern. Allein das Schreiben von Billionen von Objekten in S3 würde aufgrund der Transaktionskosten 500.000 US-Dollar kosten. Allein die Upload-Kosten für ein 100-kB-Objekt sind genauso hoch, wie bis zu zwei Monate Speichergebühr. Objekte, die kleiner als 128 kB sind, kosten das gleiche wie Objekte mit einer Größe von 128 kB. Diese Kostenstrukturen spiegeln die Effizienz des zugrunde liegenden Objektspeichers wider und sind der Grund dafür, dass Amazon Benutzern empfiehlt, mit größeren Objekten zu arbeiten.

[0005] Das Problem des ineffizienten Cloud-Speichers wird durch nicht konventionelle Datentypen, wie Nachrichtenströme und Schlüsselwerttabellen, noch verstärkt. Ein wichtiges Merkmal von Nachrichtenströmen ist, dass ein Datenstrom oft ein sehr langlebiges Objekt ist (eine Lebensdauer von Jahren ist nicht unangemessen), Aktualisierungen und Zugriffe auf den Datenstrom jedoch normalerweise während seiner gesamten Lebensdauer erfolgen. Es kann wünschenswert sein, dass ein Dateiserver einen Teil des Datenstroms in einen Cloud-Dienst eines Drittanbieters verlagert, um Speicherplatz zu sparen, aber ein Teil des Datenstroms bleibt jedoch möglicherweise aktiv und wird daher häufig von den Dateiserverprozessen verwendet. Dies bedeutet häufig, dass nur kleine zusätzliche Teile eines Nachrichtenstroms jeweils an die kalte Stufe gesendet werden können, während ein Großteil des Objekts auf dem Dateiserver gespeichert bleibt. Lösungsansätze hierzu sind beispielsweise aus den Druckschriften US 2014/0 379 715 A1 und US 2016/0 041 907 A1 bekannt.

[0006] Sicherheit ist auch eine Schlüsselanforderung für jedes System, das kalte Daten in einem Cloud-Dienst speichert. Die Erfindung macht es sich zur Aufgabe, die oben beschriebenen Nachteile zumindest abzumildern. Diese Aufgabe wird durch die in den unabhängigen Ansprüchen angegebene Erfindung gelöst.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0007] Eine oder mehrere Ausführungsformen der vorliegenden Offenbarung sind beispielhaft und nicht einschränkend in den Figuren der beigefügten Zeichnungen dargestellt, in denen gleiche Bezugszeichen ähnliche Elemente angeben.

Fig. 1A zeigt ein Blockdiagramm, das eine Umgebung zum Implementieren eines mehrstufigen Dateispeichersystems gemäß einer Ausführungsform darstellt.

Fig. 1 B zeigt ein schematisches Diagramm, das logische Organisationen von Daten im Dateisystem darstellt.

Fig. 2A zeigt ein Beispiel von Schnappschüssen eines Datenvolumens.

Fig. 2B zeigt ein Blockdiagramm, das Prozesse zum Auslagern von Daten in eine kalte Stufe darstellt.

Fig. 3 zeigt ein Blockdiagramm, das Elemente und Kommunikationspfade in einer Leseoperation in einem mehrstufigen Dateisystem gemäß einer Ausführungsform darstellt.

Fig. 4 zeigt ein Blockdiagramm, das Elemente und Kommunikationspfade in einer Schreiboperation in einem mehrstufigen Dateisystem gemäß einer Ausführungsform darstellt.

Fig. 5 zeigt ein Blockdiagramm eines Computersystems, wie es verwendet werden kann, um bestimmte Merkmale einiger der Ausführungsformen zu implementieren.

DETAILLIERTE BESCHREIBUNG

[0008] Verschiedene beispielhafte Ausführungsformen werden nun beschrieben. Die folgende Beschreibung enthält bestimmte spezifische Details, um diese Beispiele gründlich zu verstehen und beschreiben zu können. Ein Fachmann der relevanten Technologie wird jedoch verstehen, dass einige der offenbarten Ausführungsformen ohne viele dieser Details ausgeführt werden können.

[0009] Ebenso wird ein Fachmann der relevanten Technologie verstehen, dass einige der Ausführungsformen viele andere offensichtliche Merkmale enthalten können, die hier nicht im Detail beschrieben sind. Außerdem werden einige bekannte Strukturen oder Funktionen möglicherweise im Folgenden nicht detailliert gezeigt oder beschrieben, um zu vermeiden, dass die relevanten Beschreibungen der verschiedenen Beispiele unnötig verdeckt werden.

[0010] Die nachstehend verwendete Terminologie ist in ihrem weitesten sinnvollen Umfang zu interpretieren, obwohl sie in Verbindung mit einer detaillierten Beschreibung bestimmter spezifischer Beispiele der Ausführungsformen verwendet wird. In der Tat können im Folgenden sogar bestimmte Begriffe hervorgehoben werden. Alle Begriffe, die auf eine beschränkte Weise interpretiert werden sollen, werden jedoch offen und spezifisch als solche in der detaillierten Beschreibung definiert.

Systemübersicht

[0011] Ein mehrstufiges Dateispeichersystem bietet eine richtlinienbasierte automatisierte Einstufungsfunktion, die sowohl ein Dateisystem mit vollständiger Lese-/Schreibsemantik als auch einen Cloud-basierten Objektspeicher von Drittanbietern als zusätzliche Speicherstufe verwendet. Das mehrstufige Dateispeichersystem verwendet einen Dateiserver (z. B. intern von einem Unternehmen betrieben) in Kommunikation mit entfernten Servern von Drittanbietern, um verschiedene Arten von Daten zu verwalten. In einigen Ausführungsformen empfängt der Dateiserver eine Datenanforderung von einem Benutzergerät. Die Daten werden auf dem Dateiserver durch einen virtuellen Cluster-Deskriptor dargestellt. Der Dateiserver fragt eine Kennungszuordnung unter Verwendung einer Kennung des virtuellen Cluster-Deskriptors ab. Als Reaktion auf die Angabe durch die Kennungszuordnung, dass die angeforderten Daten an einem vom Dateiserver entfernten Ort gespeichert sind, greift der Dateiserver auf eine Übersetzungstabelle von kalten Stufen zu, in der eine Zuordnung zwischen einer Kennung eines jeden von mehreren virtuellen Cluster-Deskriptoren und einem Speicherort von Daten gespeichert ist, die dem jeweiligen virtuellen Cluster-Deskriptor zugeordnet sind. Die Übersetzungstabelle von kalten Stufen wird unter Verwendung der Kennung des virtuellen Cluster-Deskriptors abgefragt, um einen Speicherort der angeforderten Daten zu identifizieren, und die Daten werden von dem identifizierten Speicherort auf den Dateiserver geladen.

[0012] Die Verwendung des Speichers von Drittanbietern spricht das Problem des schnellen Datenwachstum an und verbessert die Speicherressourcen von Rechenzentren, indem der Speicher von Drittanbietern als wirtschaftliche Speicherstufe mit einer enormen Kapazität für „kalte“ oder „eingefrorene“ Daten verwendet wird, auf die selten zugegriffen wird. Auf diese Weise können wertvolle lokale Speicherressourcen für aktivere Daten und Anwendungen verwendet werden, während kalte Daten zu geringeren Kosten und Verwaltungsaufwand aufbewahrt werden. Die Datenstrukturen im Dateiserver ermöglichen den Zugriff auf kalte Daten mit denselben Verfahren, die für heiße Daten („hot data“) verwendet werden.

[0013] Fig. 1A zeigt ein Blockdiagramm, das eine Umgebung zum Implementieren eines mehrstufigen Dateispeichersystems gemäß einer Ausführungsform darstellt. Wie in **Fig.** 1A gezeigt, kann die Umgebung ein Dateisystem 100 und eine oder mehrere kalte Speichervorrichtungen 150 enthalten. Das Dateisystem 100 kann ein verteiltes Dateisystem sein, das traditionelle Objekte wie Dateien, Verzeichnisse und Links sowie erstklassige Objekte wie Schlüsselwerttabellen und Nachrichtenströme unterstützt. Die kalten Speichervorrichtungen 150 können zusammen mit Speichervorrichtungen angeordnet sein, die dem Dateisystem 100 zugeordnet sind, oder die kalten Speichervorrichtungen 150 können einen oder mehrere Server umfassen, die physisch von dem Dateisystem 100 entfernt sind. Beispielsweise können die kalten Speichervorrichtungen 150 Cloud-Speichervorrichtungen sein. Von den kalten Speichervorrichtungen 150 gespeicherte Daten können in einem oder mehreren Objektpools 155 organi-

siert werden, von denen jeder eine logische Darstellung eines Datensatzes ist.

[0014] Durch das Dateisystem 100 und die kalten Speichervorrichtungen 150 gespeicherte Daten werden in eine „heiße“ Stufe und eine „kalte“ Stufe unterteilt. Im Allgemeinen handelt es sich bei „heißen“ Daten um Daten, auf die aktiv zugegriffen wird oder auf die häufig zugegriffen wird, während es sich bei „kalten“ Daten um Daten handelt, auf die voraussichtlich nur selten zugegriffen wird. Beispielsweise können kalte Daten solche Daten enthalten, die für behördliche oder Compliance-Zwecke aufbewahrt werden müssen. Speichervorrichtungen, die dem Dateisystem 100 zugeordnet sind, bilden die heiße Stufe, die die heißen Daten speichert. Das lokale Speichern der heißen Daten im Dateisystem 100 ermöglicht es dem Dateisystem 100, schnell auf die heißen Daten zuzugreifen, wenn dies angefordert wird, und bietet schnelle Antworten auf Datenanforderungen mit geringeren Verarbeitungskosten als beim Zugriff auf die kalte Stufe. Die kalten Speichervorrichtungen 150 können die kalten Daten speichern und die kalte Stufe bilden. Durch das Offload oder Auslagern selten verwendeter Daten in die kalte Stufe wird Speicherplatz im Dateisystem 100 für neue Daten freigegeben. Das Abrufen von Daten aus der kalten Stufe kann jedoch erheblich kostspieliger und zeitintensiver sein als der Zugriff auf lokal gespeicherte Daten.

[0015] Daten können basierend auf Regeln und Richtlinien, die von einem Administrator des Dateisystems 100 festgelegt wurden, als heiß oder kalt identifiziert werden. Diese Regeln können beispielsweise die Zeit seit dem letzten Zugriff, seit der Änderung oder seit der Erstellung umfassen. Regeln können für verschiedene Datentypen variieren (z. B. können Regeln, die auf eine Datei angewendet werden, sich von den Regeln unterscheiden, die auf ein Verzeichnis angewendet werden). Alle neuen Daten, die in dem Dateisystem 100 erstellt wurden, können anfänglich als heiße Daten klassifiziert und in eine lokale Speichervorrichtung in dem Dateisystem 100 geschrieben werden. Sobald Daten als kalt klassifiziert wurden, werden sie in die kalte Stufe ausgelagert. Das Lesen und Schreiben kalter Daten kann ein teilweises Zwischenspeichern oder ein anderes temporäres Speichern der Daten lokal im Dateisystem 100 verursachen. Ausgelagerte Daten können jedoch nicht erneut als „heiß“ klassifiziert werden, wenn keine Verwaltungsaktion ausgeführt wird, wie beispielsweise das Ändern einer auf die Daten angewendeten Regel oder das Rückrufen eines gesamten Datenvolumens in das Dateisystem 100.

[0016] Das Dateisystem 100 verwaltet Daten, die über mehrere Clusterknoten 120 gespeichert sind, von denen jeder eine oder mehrere Speichervorrichtungen enthält. Jeder Clusterknoten 120 hostet einen oder mehrere Speicherpools 125. In jedem Speicherpool 125 sind Daten in Containern 127 strukturiert. Die Container 127 können Teile von Dateien, Verzeichnissen, Tabellen und Streams sowie Verknüpfungsdaten enthalten, die logische Verbindungen zwischen diesen Elementen darstellen. Jeder Container 127 kann bis zu einer bestimmten Datenmenge, beispielsweise 30 GB, aufnehmen, und jeder Container 127 kann vollständig in einem der Speicherpools 125 enthalten sein. Die Container 127 können auf einen anderen Clusterknoten 120 repliziert werden, wobei ein Container als Master bezeichnet wird. Beispielsweise kann der Container 127A ein Master-Container für bestimmte darin gespeicherte Daten sein, und der Container 127D kann ein Replikat der Daten speichern. Die Container 127 und die logische Darstellung der von den Containern bereitgestellten Daten sind für Endbenutzer des Dateisystems 100 möglicherweise nicht sichtbar.

[0017] Wenn Daten in einen Container 127 geschrieben werden, werden die Daten auch in jeden Container 127 geschrieben, der ein Replikat der Daten enthält, bevor das Schreiben bestätigt wird. In einigen Ausführungsformen werden Daten, die in einen Container 127 geschrieben werden sollen, zuerst an den Master-Container gesendet, der seinerseits die Schreibdaten an die anderen Replikate sendet. Wenn ein Replikat einen Schreibvorgang nicht innerhalb eines bestimmten Zeitraums und nach einer festgelegten Anzahl von Wiederholungsversuchen bestätigt, kann die Replikatkette für den Container 127 aktualisiert werden. Ein dem Container 127 zugeordneter Epochenzähler kann ebenfalls aktualisiert werden. Der Epochenzähler ermöglicht es jedem Container 127, zu überprüfen, ob zu schreibende Daten aktuell sind, und veraltete Schreibvorgänge von Master-Containern früherer Epochen abzulehnen.

[0018] Wenn sich ein Speicherpool 125 von einem vorübergehenden Fehler erholt, sind die Container 127 im Pool 125 möglicherweise nicht so veraltet. Als solches kann das Dateisystem 100 eine Kulanzfrist anwenden, nachdem der Verlust eines Container-Replikats festgestellt wurde, bevor ein neues Replikat erstellt wird. Wenn das verlorene Replikat eines Containers vor Ablauf der Kulanzfrist wiedererscheint, kann es erneut mit dem aktuellen Status des Containers synchronisiert werden. Sobald das Replikat aktualisiert wurde, wird die Epoche für den Container inkrementiert und das neue Replikat der Replikationskette für den Container hinzugefügt.

[0019] Innerhalb eines Containers 127 können Daten in Blöcke segmentiert und in einer Datenstruktur wie einem B-Baum organisiert werden. Die Datenblöcke enthalten bis zu einer spezifizierten Datenmenge (z. B. 8 kB) und können in Gruppen einer spezifizierten Anzahl von Blöcken (z. B. 8) kom-

primiert werden. Wenn eine Gruppe komprimiert ist, kann die Aktualisierung eines Blocks das Lesen und Schreiben mehrerer Blöcke aus der Gruppe umfassen. Wenn die Daten nicht komprimiert sind, kann jeder einzelne Block direkt überschrieben werden.

[0020] In dem Dateisystem 100 gespeicherte Daten können Endbenutzern als Datenträger dargestellt werden. Jedes Volume kann einen oder mehrere Container 127 enthalten. Bei der Darstellung für einen Endbenutzer kann ein Volume ein ähnliches Erscheinungsbild wie ein Verzeichnis haben, jedoch zusätzliche Verwaltungsfunktionen enthalten. Jedes Volume kann einen Einhängepunkt haben, der einen Ort in einem Namespace definiert, an dem das Volume sichtbar ist. Operationen in dem Dateisystem 100 zum Verarbeiten von kalt eingestuften Daten, wie zum Beispiel Schnappschuss-Erstellung, Spiegelung und lokales Definieren von Daten innerhalb eines Clusters, können auf Volume-Ebene ausgeführt werden.

[0021] Das Dateisystem 100 enthält ferner eine Containerstandortdatenbank (CLDB) 110. Die CLDB 110 führt Informationen darüber, wo sich jeder Container 127 befindet, und erstellt die Struktur jeder Replikationskette für Daten, die von dem Dateisystem 100 gespeichert werden. Die CLDB 110 kann von mehreren redundanten Servern geführt werden und Daten in der CLDB können selbst in Containern 127 gespeichert werden. Dementsprechend kann die CLDB 110 auf ähnliche Weise wie andere Daten im Dateisystem 100 repliziert werden, so dass die CLDB mehrere Hot-Standbys haben kann, die im Falle eines CLDB-Fehlers übernommen werden können. Die Bestimmung einer Master-CLDB 110 kann unter Verwendung einer auf einem Koordinierungsdienst basierenden Führerwahl erfolgen. In einer Ausführungsform verwendet der Koordinierungsdienst Apache Zookeeper, um konsistente Aktualisierungen bei Knotenausfällen oder Netzwerkpartitionen sicherzustellen.

[0022] Die CLDB 110 kann Eigenschaften und Regeln speichern, die sich auf Einstufungsdienste beziehen. Zum Beispiel kann die CLDB 110 Regeln speichern, um selektiv Daten zu identifizieren, die in die kalte Stufe ausgelagert werden sollen, und um zu planen, wann Daten ausgelagert werden sollen. Die CLDB 110 kann auch Objektpooleigenschaften speichern, die zum Speichern von und Zugreifen auf ausgelagerte Daten verwendet werden. Beispielsweise kann die CLDB 110 eine IP-Adresse des Speichergeräts speichern, auf dem abgeladene Daten, Authentifizierungsdaten für den Zugriff auf das Speichergerät, die Komprimierungsstufe, Verschlüsselungsdetails oder empfohlene Objektgrößen gespeichert sind.

[0023] Zusammenfassend wird der Begriff „Einstufungsdienste“ hier verwendet, um sich auf verschiedene unabhängige Dienste zu beziehen, die verschiedene Aspekte des Datenlebenszyklus für eine bestimmte Einstufungsebene verwalten. Diese Dienste werden in der CLDB 110 für jede Stufe konfiguriert, die auf jedem Volume aktiviert ist. Die CLDB 110 verwaltet die Ermittlung, Verfügbarkeit und einen gewissen globalen Status dieser Dienste. Die CLDB 110 kann auch alle Volumes verwalten, die von diesen Diensten zum Speichern ihrer privaten Daten (z. B. Metadaten für die Dienste auf Einstufungsebene) benötigt werden, sowie alle dienstspezifischen Konfigurationen, z. B. auf welchen Hosts diese Dienste ausgeführt werden können. Im Fall einer kalten Einstufung unter Verwendung der Objektpools 155 können die Einstufungsdienste auch über bestimmte Hosts im Cluster als Gateway zum Objektpool 155 fungieren, da möglicherweise nicht alle Hosts Zugriff auf die kalten Speichervorrichtungen 150 haben.

[0024] Wie oben beschrieben, werden Daten in dem Dateisystem 100 und den kalten Speichervorrichtungen 150 in Blöcken gespeichert. **Fig.** 1 B zeigt ein schematisches Diagramm, das logische Organisationen von Daten in dem Dateisystem 100 darstellt. Wie in **Fig.** 1B gezeigt, können Datenblöcke 167 logisch in virtuelle Cluster-Deskriptoren (VCDs) 165 gruppiert werden. Beispielsweise kann jede VCD 165 bis zu acht Datenblöcke enthalten. Eine oder mehrere VCDs 165 können zusammen Daten in einem diskreten Datenobjekt darstellen, das von dem Dateisystem 100 gespeichert wird, beispielsweise eine Datei. Die VCD-(165)-Darstellung erzeugt eine Indirektionsebene zwischen der zugrunde liegenden physikalischen Speicherung von Daten und Vorgängen auf höherer Ebene im mehrstufigen Speichersystem, die Daten erstellen, lesen, schreiben, ändern und löschen. Diese übergeordneten Vorgänge können beispielsweise Lesen, Schreiben, Erstellen von Schnappschüssen, Replizieren, Neusynchronisieren und Spiegeln umfassen. Durch die Indirektion können diese Vorgänge mit der VCD-Abstraktion weiterarbeiten, ohne dass sie wissen müssen, wie oder wo die zur VCD gehörenden Daten physikalisch gespeichert sind. In einigen Ausführungsformen kann die Abstraktion nur für inhaltliche Daten gelten, die in dem mehrstufigen Speichersystem gespeichert sind; Dateisystem-Metadaten (wie z. B. Namespace-Metadaten, Inode-Listen und Fidmap) können dauerhaft auf dem Dateiserver 100 gespeichert werden, und dementsprechend kann das Dateisystem 100 möglicherweise nicht von der Abstraktion des Speicherorts der Metadaten profitieren. In anderen Fällen können die Dateimetadaten jedoch auch durch VCDs dargestellt werden.

[0025] Jedem VCD 165 ist eine eindeutige Kennung zugeordnet (hier als VCDID bezeichnet). Das Dateisystem 100 verwaltet eine oder mehrere Zuordnun-

gen 160 (hier als VCDID-Zuordnung bezeichnet), die den physikalischen Ort von Daten speichern, die jeder VCDID zugeordnet sind. Beispielsweise kann jeder Container 127 eine entsprechende VCDID-Zuordnung 160 aufweisen. In dem trivialen Fall, in dem Daten noch nicht in einen Objektpool 155 ausgelagert wurden, kann die VCDID-Zuordnung 160 eine Eins-zu-Eins-Zuordnung von mehreren VCDIDs 165 zu physikalischen Blockadressen sein, an denen die jeder VCDID zugeordneten Daten gespeichert sind. Dementsprechend kann der Dateiserver 100, wenn Daten lokal auf dem Dateiserver 100 gespeichert sind, eine VCDID-Zuordnung 160 unter Verwendung einer VCDID abfragen, um den physikalischen Ort der Daten zu identifizieren. Sobald Daten in einen Objektpool ausgelagert wurden, kann die VCDID-Zuordnung 160 leer sein oder auf andere Weise anzeigen, dass die Daten aus dem Dateisystem 100 ausgelagert wurden.

[0026] Im Allgemeinen, wenn das Dateisystem 100 eine Anforderung empfängt, die gespeicherten Daten zugeordnet ist (z. B. eine Leseanforderung oder eine Schreibanforderung), prüft das Dateisystem 100 die VCDID-Zuordnung 160 auf eine VCDID, die den angeforderten Daten zugeordnet ist. Wenn die VCDID-Zuordnung 160 eine physikalische Blockadresse für die angeforderten Daten auflistet, kann das Dateisystem 100 unter Verwendung der aufgelisteten Adresse auf die Daten zugreifen und die Datenanforderung direkt erfüllen. Wenn der Eintrag leer ist oder die VCDID-Zuordnung 160 andernfalls anzeigt, dass die Daten ausgelagert wurden, kann das Dateisystem 100 eine Sequenz von kalten Einstufungsdiensten abfragen, um die der VCDID zugeordneten Daten zu finden. Die kalten Einstufungsdienste können in einer Prioritätsreihenfolge angeordnet werden, so dass eine Löschcodierung beispielsweise der Cloud-Speicherung vorgezogen werden kann. Durch die Verwendung einer priorisierten Suche nach Einstufungsdiensten können Daten auch in mehreren Stufen (z. B. einer heißen Stufe und einer kalten Stufe) verfügbar sein, was einen Prozess zum Verschieben von Daten zwischen Stufen vereinfacht.

[0027] Das Verwenden und Führen der VCDID-Zuordnung kann die Datenabrufleistung des Dateisystems 100 auf zwei primäre Arten beeinflussen. Das Abfragen der VCDID-Zuordnung, um lokale Speicherorte für die Daten in einer VCD zu finden, erzeugt zunächst einen zusätzlichen Suchschritt, der z. B. über das Abfragen eines Datei-B-Baums hinausgeht. Dieser zusätzliche Suchschritt verursacht dem Dateisystem 100 Kosten, die hauptsächlich durch das Laden eines Caches der VCDID-Zuordnungseinträge verursacht werden. Das Verhältnis der Größe der tatsächlichen Daten in einem Container zur VCDID-Zuordnung selbst ist jedoch so groß, dass die zu amortisierenden Kosten für das Laden der Zuordnung gering sind. Darüber hinaus können Volumes mit kurzlebigen, sehr heißen Daten diese Kosten vollständig vermeiden, da die Einstufung selektiv für einige Volumes und für andere nicht aktiviert wird.

[0028] Die zweite Art von Leistungseinbußen wird durch Interferenzen zwischen Dateisystemvorgängen im Hintergrund und E/A-Vorgängen im Vordergrund verursacht. Insbesondere können Einfügungen in die VCDID-Zuordnung beim Verschieben von Daten Zeit und Verarbeitungsressourcen des Dateisystems 100 kosten. In einigen Ausführungsformen können die Kosten von Einfügungen reduziert werden, indem eine Technik verwendet wird, die einem Log-Structured-Merge-(LSM)-Baum ähnlich ist. Während ein Bereinigungsvorgang Daten verschiebt, hängt der Bereiniger neue Einträge an eine Protokolldatei an und schreibt sie in eine speicherinterne Datenstruktur. Wenn genügend Einträge im Protokoll erfasst wurden, können diese Einträge sortiert und mit dem B-Baum zusammengeführt werden, sodass geringere Amortisationskosten als beim Ausführen einzelner Einfügungen verursacht werden. Die Zusammenführung kann mit geringem Konflikt mit dem Haupt-E/A-Pfad durchgeführt werden, da Mutationen des B-Baums, der die VCDID-Zuordnung enthält, in das Nur-Anhängen-Protokoll gezwungen werden können, wodurch alle tatsächlichen Mutationen bis zum Zusammenführungsschritt verzögert werden. Die Zusammenführung des B-Baums mit den Nur-Anhängen-Protokollen kann durch einen Kompaktierungsprozess erfolgen. Obwohl diese Zusammenführungsschritte Verarbeitungsressourcen des Dateisystems 100 verbrauchen, verringert das Verschieben dieser Vorgänge aus dem kritischen E/A-Pfad die Auswirkung auf die Leistung des Dateisystems 100.

Auslagern von Daten in eine kalte Stufe

[0029] Datenoperationen im mehrstufigen Dateisystem können auf Volume-Ebene konfiguriert werden. Diese Operationen können zum Beispiel das Replizieren und Spiegeln von Daten innerhalb des Dateisystems 100 sowie Einstufungsdienste wie das kalte Einstufen unter Verwendung von Objektpools 155 umfassen. Der Administrator kann verschiedene Einstufungsdienste auf demselben Volume konfigurieren, ebenso wie mehrere Spiegel unabhängig voneinander definiert werden können.

[0030] Aus der Sicht eines Benutzers sieht eine Datei wie die kleinste logische Einheit von Benutzerdaten aus, die für das Auslagern in die kalte Stufe identifiziert wurde, da sich für ein Volume definierte Auslagerungsregeln auf Eigenschaften auf Dateiebene beziehen. Das Auslagern von Daten auf Dateibasis hat jedoch den Nachteil, dass Schnappschüsse unveränderte Daten auf physikalischer

Blockebene im Dateisystem 100 teilen. Auf diese Weise kann dieselbe Datei über Schnappschüsse hinweg viele Blöcke miteinander teilen. Das Auslagern auf Dateiebene würde dementsprechend zu einer Duplizierung der geteilten Daten in einer Datei für jeden Schnappschuss führen. Schnappschüsse auf VCD-Ebene können jedoch die geteilten Daten nutzen, um Speicherplatz zu sparen.

[0031] **Fig.** 2A zeigt ein Beispiel von Schnappschüssen eines Datenvolumens. In **Fig.** 2A werden Datenblöcke in einer Datei zwischen Schnappschüssen und der letzten beschreibbaren Ansicht der Daten geteilt. Die Beispieldatei durchläuft die folgende Abfolge von Ereignissen:

1. Die ersten 192 KB der Datei (dargestellt durch drei VCDs) werden geschrieben,

2. Schnappschuss S1 wird erstellt.

3. Die letzten 128 KB der Datei (dargestellt durch zwei VCDs) werden überschrieben.

4. Schnappschuss S2 wird erstellt.

5. Die letzten 64 kB der Datei (dargestellt durch eine VCD) werden überschrieben.

[0032] Falls die Blöcke in Schnappschuss S1 in die kalte Speichervorrichtung 150 verschoben werden, können Schnappschuss S2 und die aktuelle Version der Datei die eingestuften Daten mit Schnappschuss S1 teilen. Umgekehrt würde das Auslagern auf Dateiebene nicht die mögliche Platzersparnis von geteilten Blöcken nutzen. Dieser verschwendete Speicherplatz kann erhebliche Auswirkungen auf die Effizienz und die Kosten der Datenführung in der kalten Stufe haben, insbesondere bei langlebigen Schnappschüssen oder einer großen Anzahl von Schnappschüssen.

[0033] Wie in **Fig.** 2A gezeigt, werden Datenblöcke in einer Datei zwischen Schnappschüssen und der letzten beschreibbaren Ansicht der Daten geteilt. Wenn Datenblöcke überschrieben werden, schatten die neuen Blöcke die Blöcke in älteren Schnappschüssen ab, werden jedoch mit neueren Ansichten geteilt. Hierbei wurde der mit Offset 0 beginnende Block nie überschrieben, die mit 64k und 128k beginnenden Blöcke wurden vor der Aufnahme von Schnappschuss 2 überschrieben und der Block mit 128k wurde nach Schnappschuss 2 einige Zeit später erneut überschrieben.

[0034] Falls die in **Fig.** 2A gezeigten Daten auf Dateiebene ausgelagert wurden, muss die gesamte Datei entweder „heiß“ (im lokalen Speicher verfügbar) oder „kalt“ (im Objektpool gespeichert) sein, und Remote-E/A-Vorgänge für Dateien in Datenblöcken sind viel schwieriger zu verwalten. Da einige Datentypen, z. B. Nachrichtenströme, sowohl sehr heiße als auch sehr kalte Daten im selben Objekt enthalten können, ist es ineffizient, zu bestimmen, ob das gesamte Objekt lokal oder in der kalten Stufe gespeichert werden soll. Im Gegensatz dazu ermöglicht die Einstufung auf der Cluster-Deskriptor-Ebene dem Dateisystem 100, Daten effizienter zu klassifizieren. In Bezug auf die Datenblöcke in **Fig.** 2A können alle Blöcke in den Schnappschüssen 1 und 2 als kalt betrachtet werden, während das Dateisystem 100 den eindeutigen Block der neuesten Version als heiße Daten beibehält.

[0035] **Fig.** 2B zeigt ein Blockdiagramm, das Prozesse zum Auslagern von Daten in eine kalte Stufe darstellt. Wie in **Fig.** 2B gezeigt, können die Prozesse einen Übersetzer von kalten Stufen 202, einen Offloader von kalten Stufen 205 und einen Kompaktor von kalten Stufen 204 umfassen. Der Übersetzer von kalten Stufen 202, der Offloader von kalten Stufen 205 und der Kompaktor von kalten Stufen 204 können jeweils von einem oder mehreren Prozessoren des Dateisystems 100 ausgeführt und als Softwaremodule, Hardwaremodule oder eine Kombination von diesen konfiguriert werden. Alternativ kann jeder der Prozesse von einer Rechenvorrichtung ausgeführt werden, die sich vom Dateisystem 100 unterscheidet, aber vom Dateisystem 100 aufgerufen werden kann.

[0036] Der Übersetzer von kalten Stufen (CTT) 202 ruft Daten aus dem Objektpool 155 ab, der einer gegebenen VCDID zugeordnet ist. Um dies zu erreichen, führt der CTT 202 interne Datenbanktabellen 203, die die VCDIDs in einen Ort einer entsprechenden VCD übersetzen, wobei der Ort als eine Objektkennung und ein Offset zurückgegeben wird. Sie kann auch alle erforderlichen Informationen speichern, um die aus dem Objektpool 155 abgerufenen Daten (z. B. einen Hash oder eine Prüfsumme) zu validieren, die Daten zu dekomprimieren, falls der Komprimierungsgrad zwischen dem Objektpool 155 und dem Dateisystem 100 unterschiedlich ist, und um sie zu entschlüsseln, falls eine Verschlüsselung aktiviert ist. Wenn Daten in den Objektpool 155 ausgelagert werden, können die CTT-Tabellen 203 mit einem Eintrag für die VCDIDs aktualisiert werden, die den ausgelagerten Daten entsprechen. Der CTT 202 kann auch die Tabellen 203 nach einer Neukonfiguration der Objekte in dem Objektpool 155 aktualisieren. Eine beispielhafte Objekt-Rekonfigurierung ist die Kompaktierung des Objektpools 155 durch den nachstehend beschriebenen Kompaktor von kalten Stufen 204. Der CTT 202 kann ein beständiger Prozess sein, und da jeder Containerprozess den Ort des CTT 202 kennen kann, kann das Dateisystem 100 zu jeder Zeit Daten für beliebige VCDIDs anfordern. Um zu wissen, wo ein CTT-Prozess ausgeführt wird, kann das Dateisystem 100 Kontaktinformationen wie IP-Adresse und Portnummer in der CLDB 110 speichern. Alternativ kann das Dateisystem 100 die Kontaktinformationen des CTT 202 spei-

chern, nachdem es von ihm kontaktiert wurde. Eine weitere Alternative besteht darin, dass der Dateisystemprozess eine Verbindung zum CTT 202 aufrechterhält, nachdem die Verbindung entweder vom CTT 202 oder vom Dateisystemprozess geöffnet wurde.

[0037] Der Offloader von kalten Stufen (CTO) 205 identifiziert Dateien in dem Volume, die zum Auslagern bereit sind, holt Daten, die diesen Dateien entsprechen, aus dem Dateisystem 100 und packt diese Daten in Objekte, die in einen Objektpool 155 geschrieben werden sollen. Der CTO-(205)-Prozess kann nach einem definierten Zeitplan gestartet werden, der in der CLDB 110 konfiguriert werden kann. Um auszulagernde Dateien zu identifizieren, kann der CTO 205 Informationen 207 darüber abrufen, welche Container 127 sich in einem Volume befinden, und dann 208 Listen von Inodes und Attributen für diese Container aus dem Dateisystem 100 abrufen. Der CTO 205 kann die Volume-spezifischen Einstufungsregeln auf diese Informationen anwenden und Dateien oder Teile von Dateien identifizieren, die die Anforderungen für den Wechsel zu einer neuen Stufe erfüllen. Die so identifizierten Daten können eine Anzahl von Seitenclustern (z. B. in Schritten von 64 kB) umfassen, die zu vielen Dateien gehören. Diese Seitencluster können gelesen 209 und zusammengepackt werden, um ein Objekt zum Einstufen zu bilden, das beispielsweise eine Größe von 8 MB oder mehr aufweisen kann. Während des Packens von Daten in die Objekte, berechnet der CTO 205 Validierungsdaten (wie einen Hash oder eine Prüfsumme), die später zur Konsistenzprüfung verwendet werden können, komprimiert die Daten bei Bedarf und verschlüsselt die Daten bei Bedarf. Das resultierende Objekt wird in die kalte Stufe 211 geschrieben 210 (z. B. zur Speicherung an eine kalte Speichervorrichtung 150 gesendet). Der CTO stellt sicher 212, dass die VCDID-Zuordnungen in den internen CTT-Tabellen 203 aktualisiert werden, bevor er das Dateisystem 100 benachrichtigt 213, die VCDID in seiner lokalen VCDID-Zuordnung als ausgelagert zu markieren.

[0038] Der Kompaktor von kalten Stufen (CTC) 204 identifiziert gelöschte VCDIDs und entfernt sie aus den CTT-Tabellen 203. Vorgänge wie Löschen von Dateien, Löschen von Schnappschüssen und Überschreiben vorhandener Daten können das logische Entfernen von Daten im Dateisystem 100 bewirken. Letztendlich führen diese Operationen zum Löschen von VCDIDs aus den VCDID-Zuordnungen. Um gelöschte VCDIDs zu entfernen, untersucht 214 der CTC 204 die VCDID-Zuordnung, um Möglichkeiten zum vollständigen Löschen oder Kompaktieren 215 von Objekten zu finden, die in den kalten Pools gespeichert sind. Ferner kann der CTC-(204)-Dienst auch ungültige Daten in Objekten verfolgen, die sich im Objektpool befinden, und Objekte löschen, die im Laufe der Zeit ungültig geworden sind, wodurch Speicherplatz im Objektpool frei wird. Zufällige Löschvorgänge können jedoch zu einer Fragmentierung von Daten führen, die zu ungenutztem Speicherplatz in den Objekten im Objektpool führt. Dementsprechend kann der CTC-Dienst 204 gelöschte Objekte entfernen, während die Menge an nicht verwendetem Speicherplatz kleiner als ein Schwellenwert bleibt. Dieser Dienst kann auch Speicherplatz von solchen defragmentierten Objekten abrufen, indem Objekte mit großem, nicht verwendetem Speicherplatz in neue Objekte kompaktiert und Zuordnungen in der CTT 202 aktualisiert werden. Der CTC 204 kann in geplanten Intervallen ausgeführt werden, was in der CLDB 110 konfiguriert werden kann.

[0039] Der vom CTC 204 durchgeführte Kompaktierungsprozess kann trotz Aktualisierungen der Daten im Dateisystem sicher fortgesetzt werden. Da die VCDID-Zuordnung und jeder kalte Pool nacheinander geprüft werden, kann das Hinzufügen eines Verweises in der VCDID-Zuordnung für einen bestimmten Block dazu führen, dass Änderungen in den nachgeordneten Einstufungsstrukturen irrelevant werden. Somit kann der CTC 204 die Einstufungsstruktur vor oder nach dem Ändern der VCDID-Zuordnung ändern, ohne die Ansicht eines Benutzers über den Zustand der Daten zu beeinträchtigen. Da darüber hinaus eingestufte Kopien von Daten unveränderlich sein können und Verweise innerhalb eines Datenblocks auf einen anderen Datenblock letztendlich über die VCDID-Zuordnung abgebildet werden, können die Daten ohne Implementierung von Überprüfungen wie verteilten Sperren sauber aktualisiert werden.

[0040] Der CTT 202, der CTO 205 und der CTC 204 können jeweils mehrere Volumes bedienen, da interne Metadaten auf Volume-Ebene getrennt sind. In einigen Ausführungsformen kann die CLDB 201 sicherstellen, dass zu einem bestimmten Zeitpunkt nur ein Dienst jedes Typs für ein bestimmtes Volume aktiv ist. Die CLDB 201 kann auch Dienste basierend auf dem Clusterstatus und den von diesen Diensten empfangenen Heartbeats stoppen oder neu starten, um eine hohe Verfügbarkeit der Einstufungsdienste sicherzustellen.

Beispiele von Operationen auf gestuften Daten

[0041] Fig. 3 zeigt ein Blockdiagramm, das Elemente und Kommunikationspfade in einer Leseoperation in einem mehrstufigen Dateisystem gemäß einer Ausführungsform darstellt. Komponenten und Prozesse, die mit Bezug auf **Fig.** 3 beschrieben werden, können denen ähnlich sein, die mit Bezug auf die **Fig.** 1 und 2B beschrieben wurden.

[0042] Wie in **Fig.** 3 gezeigt, sendet ein Client 301 eine Leseanforderung an einen Dateiserver 303. Die

Leseanforderung identifiziert vom Client 301 angeforderte Daten, beispielsweise zur Verwendung in einer vom Client 301 ausgeführten Anwendung. Der Dateiserver 303 kann einen veränderlichen Container oder ein unveränderliches Replikat gewünschter Daten enthalten. Jeder Container oder jedes Replikat ist mit einer Reihe von Verzeichnisinformationen und Dateidaten verknüpft, die beispielsweise in einem B-Baum gespeichert sind.

[0043] Der Dateiserver 303 kann den B-Baum prüfen, um die VCDID zu finden, die den angeforderten Daten entspricht, und die VCDID-Zuordnung prüfen, um den Ort der VCDID zu identifizieren. Wenn die VCDID-Zuordnung eine Liste von einer oder mehreren physikalischen Blockadressen identifiziert, an denen die Daten gespeichert sind, liest der Dateiserver 303 die Daten von dem durch die physikalischen Blockadressen angegebenen Ort, speichert die Daten in einem lokalen Cache und sendet 304 eine Antwort an den Client 301. Falls die VCDID-Zuordnung anzeigt, dass die Daten nicht lokal gespeichert sind (z. B. wenn die Zuordnung für die gegebene VCDID leer ist), identifiziert der Dateiserver 303 einen Objektpool, in den die Daten ausgelagert wurden.

[0044] Da das Abrufen der Daten aus dem Objektpool mehr Zeit in Anspruch nehmen kann als das Lesen der Daten von einer Festplatte, kann der Dateiserver 303 eine Fehlermeldung (EMOVED) an den Client 301 senden 305. Als Reaktion auf die Fehlermeldung kann der Client 301 eine nachfolgende Leseoperation 306 um ein voreingestelltes Zeitintervall verzögern. In einigen Ausführungsformen kann der Client 301 die Leseoperation 306 eine bestimmte Anzahl von Malen wiederholen. Wenn der Client 301 die Daten nach der angegebenen Anzahl von Versuchen nicht aus dem Cache des Dateiservers 303 lesen kann, gibt der Client 301 möglicherweise eine Fehlermeldung an die Anwendung zurück und unternimmt keine weiteren Versuche, die Daten zu lesen. Die Zeitspanne zwischen den Leseversuchen kann gleich sein oder sich nach jedem fehlgeschlagenen Versuch schrittweise erhöhen.

[0045] Nach dem Senden der EMOVED-Fehlermeldung an den Client 301 kann der Dateiserver 303 den Prozess des Abrufs von Daten aus der kalten Stufe beginnen. Der Dateiserver 303 kann eine Anfrage an den CTT 308 mit einer Liste von einer oder mehreren VCDIDs, die den angeforderten Daten entsprechen, senden 307.

[0046] Der CTT 308 fragt seine Übersetzungstabellen für jede der einen oder mehreren VCDIDs ab. Die Übersetzungstabellen können eine Zuordnung von den VCDIDs zu Objekt-ID und Offsets enthalten, die den Ort der entsprechenden Daten identifizieren. Unter Verwendung der Objekt-ID und des Offsets ruft der CTT 308 die Daten von der kalten Stufe 311 ab 310. Der CTT 308 validiert zurückgegebene Daten gegen einen erwarteten Wert, und wenn die erwarteten und tatsächlichen Validierungsdaten übereinstimmen, werden die Daten an den Dateiserver 303 zurückgegeben 312. Falls die gespeicherten Daten komprimiert oder verschlüsselt wurden, kann der CTT 308 die Daten dekomprimieren oder entschlüsseln, bevor die Daten an den Dateiserver 303 zurückgesendet werden 312.

[0047] Wenn der Dateiserver 303 die Daten vom CTT 308 empfängt, speichert der Dateiserver 303 die empfangenen Daten in einem lokalen Cache. Wenn eine nachfolgende Leseanforderung 306 vom Client 301 empfangen wird, sendet der Dateiserver 303 die gewünschten Daten aus dem Cache zurück 304.

[0048] **Fig.** 3 bietet einen allgemeinen Überblick über Elemente und Kommunikationspfade bei einer Leseoperation. Leseoperationen können schnell ausgeführt werden, wenn Daten lokal auf dem Dateiserver 303 gespeichert sind. Wenn die Daten nicht lokal gespeichert sind, kann der Dateiserver 303 eine Fehlermeldung an den Client 301 zurücksenden, wodurch der Client die Daten wiederholt erneut anfordert, während der Dateiserver 303 die gewünschten Daten asynchron abruft. Diese Art des Lesens vermeidet lange Anforderungen vom Client. Stattdessen wiederholt der Client Anforderungen, bis eine bestimmte Anzahl fehlgeschlagener Versuche erreicht wurde oder die gewünschten Daten empfangen wurden. Da der Client 301 die Datenanforderungen wiederholt, muss der Dateiserver 303 keine Informationen über den Status des Clients aufbewahren, während Daten von der kalten Stufe abgerufen werden. Unter Verwendung des mit Bezug auf **Fig.** 3 beschriebenen Verfahrens können viele Anfragen des Clients schnell erfüllt werden. Dies kann die Anzahl der ausstehenden Anforderungen auf der Serverseite sowie die Auswirkungen eines Dateiserverabsturzes verringern. Da in der Regel viele Clients Anforderungen an jeden Dateiserver stellen, bedeutet das Setzen eines höheren Status auf der Clientseite, dass ein höherer Status einen Dateiserverabsturz überlebt, sodass Vorgänge schneller fortgesetzt werden können.

[0049] **Fig.** 4 zeigt ein Blockdiagramm, das Elemente und Kommunikationspfade in einer Schreiboperation in einem mehrstufigen Dateisystem gemäß einer Ausführungsform darstellt. Komponenten und Prozesse, die mit Bezug auf **Fig.** 4 beschrieben werden, können denen ähnlich sein, die mit Bezug auf die **Fig.** 1, 2B und 3 beschrieben wurden.

[0050] Wie in **Fig.** 4 gezeigt, sendet ein Datei-Client 401 eine Schreibanforderung 402 an den Datei-Server 403. Die Schreibanforderung enthält eine Modifi-

kation von Daten, die von dem Dateiserver 403 oder einer entfernten Speichervorrichtung gespeichert werden, wie beispielsweise das Ändern eines Teils der gespeicherten Daten oder das Hinzufügen zu den gespeicherten Daten. Die zu ändernden Daten können auf mehrere Speichergeräte repliziert werden. Beispielsweise können die Daten sowohl auf dem Dateiserver 403 als auch auf einem oder mehreren entfernten Speichergeräten gespeichert sein oder die Daten können auf mehreren entfernten Speichergeräten gespeichert sein.

[0051] Wenn der Dateiserver 403 die Schreibanforderung vom Client 401 empfängt, kann der Dateiserver 303 den neu geschriebenen Daten eine neue VCDID zuweisen. Die neuen Daten können an beliebige andere Speichervorrichtungen 404 gesendet werden, die Replikate der zu ändernden Daten führen, so dass die anderen Server 404 die Replikate aktualisieren können.

[0052] Der Dateiserver 403 kann den B-Baum prüfen, um die VCDID der zu modifizierenden Daten abzurufen. Unter Verwendung der abgerufenen VCDID kann der Dateiserver 403 aus der VCDID-Zuordnung auf Metadaten für die VCD zugreifen. Wenn die Metadaten eine Liste von einer oder mehreren physikalischen Blockadressen enthalten, die einen Ort der zu modifizierenden Daten identifizieren, kann der Dateiserver 403 die Daten von den durch die Adressen identifizierten Orten lesen und die Daten in einen lokalen Cache schreiben. Der Dateiserver 403 kann die Daten im Cache gemäß den Anweisungen in der Schreibanforderung modifizieren. Die Schreiboperationen können auch an alle Vorrichtungen gesendet werden 406, die die Replikate der Daten speichern. Sobald die ursprünglichen Daten und Replikate aktualisiert worden sind, kann der Dateiserver 403 eine Antwort an den Client 401 senden 405, die angibt, dass der Schreibvorgang erfolgreich abgeschlossen wurde.

[0053] Wenn die Metadaten keine physikalischen Blockadressen für die zu ändernden Daten identifizieren (z. B. wenn die Zuordnung für die gegebene VCDID leer ist), identifiziert der Dateiserver 403 einen Objektpool, in den die Daten ausgelagert wurden. Da das Abrufen der Daten aus dem Objektpool mehr Zeit in Anspruch nehmen kann als das Lesen der Daten von einer Festplatte, kann der Dateiserver 403 eine Fehlermeldung (EMOVED) an den Client 401 senden 407. Als Reaktion auf die Fehlermeldung kann der Client 401 eine nachfolgende Schreiboperation 408 um ein voreingestelltes Zeitintervall verzögern. In einigen Ausführungsformen kann der Client 401 die Schreiboperation 408 eine bestimmte Anzahl von Malen wiederholen. Wenn der Schreibvorgang nach der angegebenen Anzahl von Versuchen fehlschlägt, gibt der Client 401 möglicherweise eine Fehlermeldung an die Anwendung zurück und versucht möglicherweise nicht erneut, die Daten zu schreiben. Die Zeitspanne zwischen den Schreibversuchen kann gleich sein oder sich nach jedem fehlgeschlagenen Versuch schrittweise erhöhen.

[0054] Nach dem Senden der EMOVED-Fehlermeldung an den Client 401 kann der Dateiserver 403 den Prozess des Abrufens von Daten von der kalten Stufe beginnen, um die Daten zu aktualisieren. Der Dateiserver 403 kann eine Anfrage 409 mit einer Liste von einer oder mehreren VCDIDs, die den zu modifizierenden Daten entsprechen, an den CTT 410 senden.

[0055] Der CTT 410 durchsucht seine Übersetzungstabellen nach der einen oder den mehreren VCDIDs und ruft unter Verwendung der von den Übersetzungstabellen ausgegebenen Objekt-ID und des Offsets die Daten von der kalten Stufe 412 ab 411. Der CTT 410 validiert die zurückgegebenen Daten gegen einen erwarteten Wert, und wenn die erwarteten und tatsächlichen Validierungsdaten übereinstimmen, werden die Daten an den Dateiserver 403 zurückgesendet 413. Wenn die gespeicherten Daten komprimiert oder verschlüsselt wurden, kann der CTT 410 die Daten dekomprimieren oder entschlüsseln, bevor die Daten an den Dateiserver 403 zurückgesendet werden 413.

[0056] Wenn der Dateiserver 403 die Daten von der CTT 410 empfängt, repliziert der Dateiserver 403 die unveränderten Daten auf beliebige Replikate und schreibt die Daten unter Verwendung derselben VCDID in einen lokalen Cache (wobei die Daten zurück in heiße Daten konvertiert werden). Wenn eine nachfolgende Schreibanforderung vom Client 401 empfangen wird, kann der Dateiserver 403 ein Überschreiben der abgerufenen Daten durchführen, um die Daten gemäß den Anweisungen in der Schreibanforderung zu aktualisieren.

[0057] Nach dem unter Bezugnahme auf **Fig.** 4 beschriebenen Prozess ist der Datenfluss der gleiche, unabhängig davon, ob die Daten lokal auf dem Dateiserver 403 gespeichert sind oder in die kalte Stufe ausgelagert wurden. Da die Schreibdaten an die Replikate gesendet werden, bevor der B-Baum überprüft wird, um den Speicherort der zu ändernden Daten zu bestimmen, müssen die Replikate die Schreibdaten möglicherweise verwerfen, wenn die zu ändernden Daten ausgelagert wurden. Obwohl dieser Prozess dazu führt, dass Daten repliziert werden, die später verworfen werden, werden die replizierten Daten nur dann verworfen, wenn die Daten ausgelagert wurden, und der Dateiserver 403 muss keine unterschiedlichen Prozesse für die Speicherung der Daten in die heiße Stufe und die Speicherung in die kalte Stufe verwenden. In anderen Ausführungsformen können jedoch die Schritte des mit Bezug auf **Fig.** 4 beschriebenen Prozesses in unter-

schiedlicher Reihenfolge durchgeführt werden. Beispielsweise kann der Dateiserver 403 den B-Baum prüfen, um den Ort der Daten zu identifizieren, bevor die Schreibanforderung an die Replikate gesendet wird.

[0058] Die Datenspeicherung in kalten Stufen unter Verwendung von Objektpools ermöglicht eine neue Option zum Erstellen schreibgeschützter Spiegel für die Notfallwiederherstellung (hier als DR-Spiegel bezeichnet). Der Objektpool wird häufig von einem Cloud-Server-Anbieter gehostet und daher auf Servern gespeichert, die physikalisch vom Dateiserver entfernt sind. Ein Volume, das in die kalte Stufe ausgelagert wurde, enthält möglicherweise nur Metadaten und zusammen mit den Metadaten, die auf dem vom kalten Einstufungsdienst verwendeten Volume gespeichert sind, machen die ausgelagerten Daten einen kleinen Bruchteil (z. B. weniger als 5 %) des tatsächlichen Speicherplatzes aus, der vom Volume verwendet wird. Ein kostengünstiger DR-Spiegel kann durch Spiegeln des Benutzer-Volumes und des vom kalten Einstufungsdienst verwendeten Volumes an einen vom Dateiserver entfernten Ort (und daher wahrscheinlich außerhalb einer den Dateiserver betreffenden Katastrophenzone) erstellt werden. Für die Wiederherstellung kann ein neuer Satz von kalten Einstufungsdiensten instanziiert werden, mit denen der DR-Spiegel nur Lesezugriff auf eine nahezu konsistente Kopie des Benutzer-Volumes hat.

Computersystem

[0059] Fig. 5 zeigt ein Blockdiagramm eines Computersystems, wie es verwendet werden kann, um bestimmte Merkmale einiger der Ausführungsformen zu implementieren. Das Computersystem kann ein Server-Computer, ein Client-Computer, ein Personal Computer (PC), ein Benutzergerät, ein Tablet-PC, ein Laptop, ein Personal Digital Assistant (PDA), ein Mobiltelefon, ein iPhone, ein iPad, ein Blackberry, ein Prozessor, ein Telefon, eine Web-Appliance, ein Netzwerk-Router, ein Switch oder eine Bridge, eine Konsole, eine tragbare Konsole, ein (tragbares) Spielgerät, ein Musik-Player, irgendein tragbares, mobiles oder tragbares Gerät oder jede Maschine sein, die in der Lage ist, sequentielle oder sonstige Anweisungen auszuführen, die von dieser Maschine auszuführende Aktionen angeben.

[0060] Das Computersystem 500 kann eine oder mehrere Zentraleinheiten („Prozessoren“) 505, einen Speicher 510, Eingabe-/Ausgabegeräte 525, z. B. Tastatur- und Zeigegeräte, Touch-Geräte, Anzeigegeräte, Speichergeräte 520, z. B. Plattenlaufwerke und Netzwerkadapter 530, z. B. Netzwerkschnittstellen, umfassen, die mit einer Verbindung 515 verbunden sind. Die Verbindung 515 ist als Abstraktion dargestellt, die einen oder mehrere separate physikalische Busse, Punkt-zu-Punkt-Verbindungen oder beides darstellt, die durch geeignete Brücken, Adapter oder Controller verbunden sind. Die Verbindung 515 kann daher beispielsweise einen Systembus, einen PCI-Bus (Peripheral Component Interconnect) oder einen PCI-Express-Bus, einen ISA-Bus (HyperTransport) oder einen SCSI-Bus (Small Computer System Interface), einen USB-Bus (Universal Serial Bus), einen IIC-Bus (12C) oder einen IEEE-Standard-1394-Bus (Institute of Electrical and Electronics Engineers) umfassen, der auch als Firewire bezeichnet wird.

[0061] Der Speicher 510 und die Speichervorrichtungen 520 sind computerlesbare Speichermedien, die Anweisungen speichern können, die zumindest Teile der verschiedenen Ausführungsformen implementieren. Zusätzlich können die Datenstrukturen und Nachrichtenstrukturen gespeichert oder über ein Datenübertragungsmedium übertragen werden, z. B. ein Signal auf einer Kommunikationsverbindung. Verschiedene Kommunikationsverbindungen können verwendet werden, z. B. das Internet, ein lokales Netzwerk, ein Weitverkehrsnetz oder eine Punkt-zu-Punkt-DFÜ-Verbindung. Somit können computerlesbare Medien computerlesbare Speichermedien enthalten, z. B. nichtflüchtige Medien und computerlesbare Übertragungsmedien.

[0062] Die im Speicher 510 gespeicherten Anweisungen können als Software und/oder Firmware implementiert werden, um den Prozessor 505 so zu programmieren, dass er die oben beschriebenen Aktionen ausführt. In einigen Ausführungsformen kann eine solche Software oder Firmware anfänglich dem Verarbeitungssystem 500 bereitgestellt werden, indem sie von einem entfernten System über das Computersystem 500 heruntergeladen wird, z. B. über Netzwerkadapter 530.

[0063] Die verschiedenen hierin eingeführten Ausführungsformen können beispielsweise durch programmierbare Schaltungen implementiert werden, z. B. einen oder mehrere Mikroprozessoren, mit Software und/oder Firmware oder vollständig in festverdrahteten (nicht programmierbaren) Spezialschaltungen oder in einer Kombination solcher Formen programmiert. Festverdrahtete Spezialschaltungen können beispielsweise in Form eines oder mehrerer ASICs, PLDs, FPGAs, usw. vorliegen.

Bemerkungen

[0064] Die obige Beschreibung und die Zeichnungen sind veranschaulichend und sollen nicht als einschränkend ausgelegt werden. Zahlreiche spezifische Details werden beschrieben, um ein gründliches Verständnis der Offenbarung zu ermöglichen. In bestimmten Fällen werden jedoch bekannte Details nicht beschrieben, um die

Beschreibung nicht zu verschleiern. Ferner können verschiedene Modifikationen vorgenommen werden, ohne vom Umfang der Ausführungsformen abzuweichen.

[0065] Ein Verweis in dieser Beschreibung auf „eine besondere Ausführungsform“ oder „eine Ausführungsform“ bedeutet, dass ein bestimmtes Merkmal, eine bestimmte Struktur oder eine bestimmte Eigenschaft, die in Verbindung mit der Ausführungsform beschrieben wurde, in mindestens einer Ausführungsform der Offenbarung enthalten ist. Das Auftreten des Ausdrucks „in einer Ausführungsform“ an verschiedenen Stellen in der Beschreibung bezieht sich nicht notwendigerweise immer auf dieselbe Ausführungsform, noch schließen separate oder alternative Ausführungsformen andere Ausführungsformen gegenseitig aus. Darüber hinaus werden verschiedene Merkmale beschrieben, die von einigen Ausführungsformen und nicht von anderen enthalten sein können. In ähnlicher Weise werden verschiedene Anforderungen beschrieben, die Anforderungen für einige Ausführungsformen sein können, jedoch nicht für andere Ausführungsformen.

[0066] Die in dieser Beschreibung verwendeten Begriffe haben im Allgemeinen ihre gewöhnliche Bedeutung im Stand der Technik, im Rahmen der Offenbarung und in dem spezifischen Kontext, in dem jeder Begriff verwendet wird. Bestimmte Begriffe, die zur Beschreibung der Offenbarung verwendet werden, wurden oben oder an anderer Stelle in der Beschreibung erörtert, um dem Praktiker zusätzliche Hinweise zur Beschreibung der Offenbarung zu geben. Der Einfachheit halber können bestimmte Begriffe hervorgehoben werden, beispielsweise durch Kursivschrift und/oder Anführungszeichen. Die Verwendung von Hervorhebungen hat keinen Einfluss auf den Umfang und die Bedeutung eines Begriffs. Der Umfang und die Bedeutung eines Begriffs sind im selben Kontext gleich, unabhängig davon, ob er hervorgehoben ist oder nicht. Es versteht sich, dass dasselbe auf mehr als eine Weise ausgedrückt werden kann.

[0067] Folglich können alternative Formulierungen und Synonyme für einen oder mehrere der hier diskutierten Begriffe verwendet werden, und ferner ist es ohne besondere Bedeutung, ob ein Begriff hier ausgearbeitet oder diskutiert wird oder nicht. Es werden Synonyme für bestimmte Begriffe bereitgestellt. Die Verwendung eines oder mehrerer Synonyme schließt die Verwendung anderer Synonyme nicht aus. Die Verwendung von Beispielen an einem beliebigen Ort in dieser Beschreibung, einschließlich Beispielen eines hierin diskutierten Begriffs, ist nur veranschaulichend und soll den Umfang und die Bedeutung der Offenbarung oder eines beispielhaften Begriffs nicht weiter einschränken. Ebenso ist die Offenbarung nicht auf die verschiedenen in dieser Beschreibung angegebenen Ausführungsformen beschränkt.

[0068] Ohne den Umfang der Offenbarung weiter einschränken zu wollen, sind Beispiele von Instrumenten, Vorrichtungen, Verfahren und deren verwandten Ergebnissen gemäß den Ausführungsformen der vorliegenden Offenbarung oben angegeben. Es ist zu beachten, dass Titel oder Untertitel in den Beispielen zur Vereinfachung des Lesens verwendet werden können, was den Umfang der Offenbarung in keiner Weise einschränken soll. Sofern nicht anders definiert, haben alle hierin verwendeten technischen und wissenschaftlichen Begriffe die gleiche Bedeutung, wie sie von einem Durchschnittsfachmann auf dem Gebiet, auf das sich diese Offenbarung bezieht, allgemein verstanden wird. Im Konfliktfall ist das vorliegende Dokument einschließlich der Definitionen maßgeblich.

Patentansprüche

1. Verfahren, umfassend:
Empfangen, an einem Dateiserver (303), von einem Benutzergerät (301), einer Anforderung nach Daten, die durch einen virtuellen Cluster-Deskriptor dargestellt werden, wobei der Dateiserver (303) eine gestuftes Datenspeichersystem (100) umfasst;
Abfragen einer Kennungszuordnung unter Verwendung einer Kennung des virtuellen Cluster-Deskriptors;
als Reaktion darauf, dass die Kennungszuordnung angibt, dass die angeforderten Daten an einem vom Dateiserver (303) entfernten Ort gespeichert sind:
Senden einer Benachrichtigung zu dem Benutzergerät (301), um das Benutzergerät (301) dazu zu veranlassen die angeforderten Daten zu einem späteren Zeitpunkt von dem Dateiserver (303) anzufordern;
Zugreifen auf eine Übersetzungstabelle für kalte Stufen (203), in der eine Zuordnung zwischen einer Kennung eines jeden von mehreren virtuellen Cluster-Deskriptoren und einem Speicherort von Daten gespeichert ist, die dem jeweiligen virtuellen Cluster-Deskriptor zugeordnet sind, wobei das gestufte Datenspeichersystem (100) einen Speicher (205) für kalte Stufen umfasst, die zum Auslagern ausgewählt sind;
Abfragen der Übersetzungstabelle (203) für kalte Stufen unter Verwendung der Kennung des virtuellen Cluster-Deskriptors, der den angeforderten Daten zugeordnet ist, um einen Speicherort der angeforderten Daten in dem Speicher (205) für kalte Stufen zu identifizieren; und
Laden der angeforderten Daten auf den Dateiserver von dem identifizierten Speicherort.

2. Verfahren nach Anspruch 1, wobei die Benachrichtigung das Benutzergerät (301) veran-

lasst, die Datenanforderung nach einer Verzögerung eines bestimmten Zeitintervalls erneut zu senden.

3. Verfahren nach Anspruch 1, ferner umfassend:
Identifizieren eines Datensatzes, der auf dem Dateiserver (303) gespeichert ist, der von dem Dateiserver an neue, vom Dateiserver entfernte Orte in dem Speicher für kalte Stufen ausgelagert werden soll, wobei der identifizierte Datensatz einem zweiten virtuellen Cluster-Deskriptor zugeordnet ist; und
Aktualisieren der Übersetzungstabelle (203) für kalte Stufen, um eine Kennung des zweiten virtuellen Cluster-Deskriptors den neuen, vom Dateiserver (303) entfernten Orten zuzuordnen.

4. Verfahren nach Anspruch 1, wobei die Kennungszuordnung eine Zuordnung zwischen einer Kennung eines virtuellen Cluster-Deskriptors und einem physikalischen Speicherort auf dem Dateiserver (303) speichert, wenn Daten, die dem virtuellen Cluster-Deskriptor entsprechen, auf dem Dateiserver gespeichert sind, und wobei die Kennungszuordnung eine Zuordnung zwischen der Kennung des virtuellen Cluster-Deskriptors und einem leeren Speicherort speichert, wenn die dem virtuellen Cluster-Deskriptor entsprechenden Daten vom Dateiserver (303) entfernt gespeichert sind.

5. Verfahren, umfassend:
Empfangen einer Anforderung nach Daten von einem Client (301), die an einem vom Dateiserver (303) entfernten, kalten Speicherort gespeichert sind, an einem Dateiserver, der ein gestuftes Datenspeichersystem umfasst, wobei der kalte Speicherort Daten umfasst, die zum Auslagern ausgewählt sind;
In Reaktion auf die Anforderung, Senden einer Benachrichtigung zu dem Benutzergerät (301), um das Benutzergerät dazu zu veranlassen die angeforderten Daten zu einem späteren Zeitpunkt von dem Dateiserver (303) anzufordern;
Zugreifen auf eine Übersetzungstabelle (203) für kalte Stufen, die eine Zuordnung zwischen einer Kennung jedes einer Vielzahl von virtuellen Cluster-Deskriptoren und einem Speicherort von Daten speichert, die dem jeweiligen virtuellen Cluster-Deskriptor zugeordnet sind;
Abfragen der Übersetzungstabelle (203) für kalte Stufen unter Verwendung einer Kennung eines virtuellen Cluster-Deskriptors, der den angeforderten Daten zugeordnet ist, um einen Speicherort der angeforderten Daten in dem kalten Speicherort zu identifizieren; und
Laden der angeforderten Daten auf den Dateiserver von dem identifizierten Speicherort.

6. Verfahren nach Anspruch 5, ferner umfassend:
Speichern auf dem Dateiserver (303) einer Kennungszuordnung, die eine Zuordnung zwischen einer Kennung eines virtuellen Cluster-Deskriptors und einem physikalischen Speicherort auf dem Dateiserver (303) speichert, wenn Daten, die dem virtuellen Cluster-Deskriptor entsprechen, auf dem Dateiserver (303) gespeichert sind, und die eine Zuordnung zwischen der Kennung des virtuellen Cluster-Deskriptors und einem leeren Speicherort speichert, wenn die Daten, die dem virtuellen Cluster-Deskriptor entsprechen, vom Dateiserver (303) entfernt gespeichert sind.

7. Verfahren nach Anspruch 6, ferner umfassend:
Abfragen der Kennungszuordnung unter Verwendung der Kennung des virtuellen Cluster-Deskriptors, der den angeforderten Daten zugeordnet ist; und
Abfragen der Übersetzungstabelle (203) für kalte Stufen als Reaktion darauf, dass die Kennungszuordnung angibt, dass die angeforderten Daten an einem vom Dateiserver (303) entfernten Ort gespeichert sind.

8. Verfahren nach Anspruch 5, wobei die Benachrichtigung den Client (301) veranlasst, die Datenanforderung nach einer Verzögerung eines bestimmten Zeitintervalls erneut zu senden.

9. Verfahren nach Anspruch 8, wobei die Benachrichtigung den Client (301) veranlasst, die Datenanforderung eine voreingestellte Anzahl von Malen erneut zu senden.

10. Verfahren nach Anspruch 5, ferner umfassend:
Identifizieren eines Datensatzes, der auf dem Dateiserver (303) gespeichert ist, der von dem Dateiserver (303) an neue vom Dateiserver (303) entfernte Orte ausgelagert werden soll, wobei der identifizierte Datensatz einem zweiten virtuellen Cluster-Deskriptor zugeordnet ist; und
Aktualisieren der Übersetzungstabelle (203) für kalte Stufen, um eine Kennung des zweiten virtuellen Cluster-Deskriptors den neuen, vom Dateiserver entfernten Orten zuzuordnen.

11. System, umfassend:
einen Prozessor (505);
ein maschinenlesbares Speichermedium (510), umfassend Anweisungen, die durch den Prozessor (505) ausführbar sind, zum Speichern von Übersetzungstabellen (203), die Kennungen von jedem einer Vielzahl von virtuellen Cluster-Deskriptoren einem physikalischen Speicherort von Daten zuordnen, die jedem virtuellen Cluster-Deskriptor entsprechen; und
einen Dateiserver (303), der ein gestuftes Dateispeichersystem (100) umfasst, welches einen kalten Speicherort zum Speichern von Daten umfasst, die

zum Auslagern ausgewählt sind, wobei der Dateiserver konfiguriert ist, zum:
Speichern einer Kennungszuordnung;
in Reaktion auf eine Datenanforderung von einem Client (301), die durch einen virtuellen Cluster-Deskriptor der Vielzahl von virtuellen Cluster-Deskriptoren repräsentiert wird, Abfragen der Kennungszuordnung unter Verwendung einer Kennung des virtuellen Cluster-Deskriptors, der mit den angeforderten Daten verknüpft ist;
als Reaktion darauf, dass die Kennungszuordnung angibt, dass die angeforderten Daten lokal auf dem Dateiserver (303) gespeichert sind, Abrufen der angeforderten Daten von dem Dateiserver (303); und
als Reaktion darauf, dass die Kennungszuordnung angibt, dass die angeforderten Daten an einem vom Dateiserver (303) entfernten Ort gespeichert sind:
Senden einer Benachrichtigung zu dem Client (301), um den Client dazu zu veranlassen, die angeforderten Daten zu einem späteren Zeitpunkt von dem Dateiserver anzufordern;
den Übersetzer von kalten Stufen unter Verwendung einer Kennung eines virtuellen Cluster-Deskriptors abzufragen, der den angeforderten Daten zugeordnet ist, um einen Speicherort der angeforderten Daten in dem kalten gestuften Speicher zu identifizieren; und
die angeforderten Daten vom identifizierten Speicherort auf den Dateiserver zu laden.

12. System nach Anspruch 11, wobei Anweisungen ausführbar sind, zum:
Identifizieren eines Satzes von Daten, die auf dem Dateiserver (303) gespeichert sind, der von dem Dateiserver an neue vom Dateiserver entfernte Orte ausgelagert werden soll, wobei der identifizierte Datensatz einem zweiten virtuellen Cluster-Deskriptor zugeordnet ist; und
Aktualisieren der Übersetzungstabelle (203) für kalte Stufen, um eine Kennung des zweiten virtuellen Cluster-Deskriptors den neuen, vom Dateiserver (303) entfernten Speicherorten zuzuordnen.

13. System nach Anspruch 11, wobei die Benachrichtigung das Benutzergerät (301) veranlasst, die Datenanforderung nach einer Verzögerung eines festgelegten Zeitraums erneut zu senden.

14. System nach Anspruch 13, wobei die Benachrichtigung das Benutzergerät (301) veranlasst, die Anforderung für die Datenanforderung eine voreingestellte Anzahl von Malen erneut zu senden.

15. System nach Anspruch 11, wobei der Dateiserver (303) ferner konfiguriert ist zum:
als Reaktion darauf, dass die Kennungszuordnung angibt, dass die angeforderten Daten lokal auf dem Dateiserver gespeichert sind, Abrufen der angeforderten Daten vom Dateiserver (303) und Bereitstellen der angeforderten Daten an das Benutzergerät (301).

16. Verfahren nach Anspruch 1, wobei die Benachrichtigung das Benutzergerät (301) dazu veranlasst, die angeforderten Daten ein weiteres Mal oder mehrere weitere Male anzufordern, während der Dateiserver (303) die angeforderten Daten asynchron aus dem kalten gestuften Speicher abruft.

17. Verfahren nach Anspruch 1, umfassend:
Nach dem Laden der angeforderten Daten auf den Dateiserver (303) von dem identifizierten Speicherort des kalten, gestuften Speichers:
Empfangen, an dem Dateiserver (303), einer weiteren Anforderung von dem Benutzergerät (301) nach den angeforderten Daten, die durch einen virtuellen Cluster-Deskriptor repräsentiert ist; und
Zurückübertragen der angeforderten Daten von dem Dateiserver (303) zu dem Benutzergerät (301) als Reaktion auf die weitere Anforderung.

18. Verfahren nach Anspruch 1, umfassend:
In Reaktion auf die Kennungszuordnung, die angibt, dass die angeforderten Daten lokal auf dem Dateiserver gespeichert sind, Abrufen der angeforderten Daten von dem Dateiserver (303) und Bereitstellen der angeforderten Daten zum Benutzergerät.

19. Verfahren nach Anspruch 5, umfassend:
nach dem Laden der angeforderten Daten von dem identifizierten Speicherort auf dem gestuften, kalten Speicher:
Empfangen, durch den Dateiserver (303), einer weiteren Anforderung von dem Client (301) für die angeforderten Daten; und
Zurückübertragen der angeforderten Daten von dem Dateiserver (303) zu dem Client (301) als Reaktion auf die weitere Anforderung.

20. System nach Anspruch 11, wobei der Dateiserver (303) dazu ausgebildet ist:
Nach dem Laden der angeforderten Daten auf den Dateiserver (303) von dem identifizierten Speicherort des kalten, gestuften Speichers:
In Reaktion auf das Empfangen einer weiteren Anforderung von dem Client nach den angeforderten Daten, die angeforderten Daten
von dem Dateiserver (303) zu dem Client als Reaktion auf die weitere Anforderung zurückzuübertragen.

Es folgen 5 Seiten Zeichnungen

Anhängende Zeichnungen

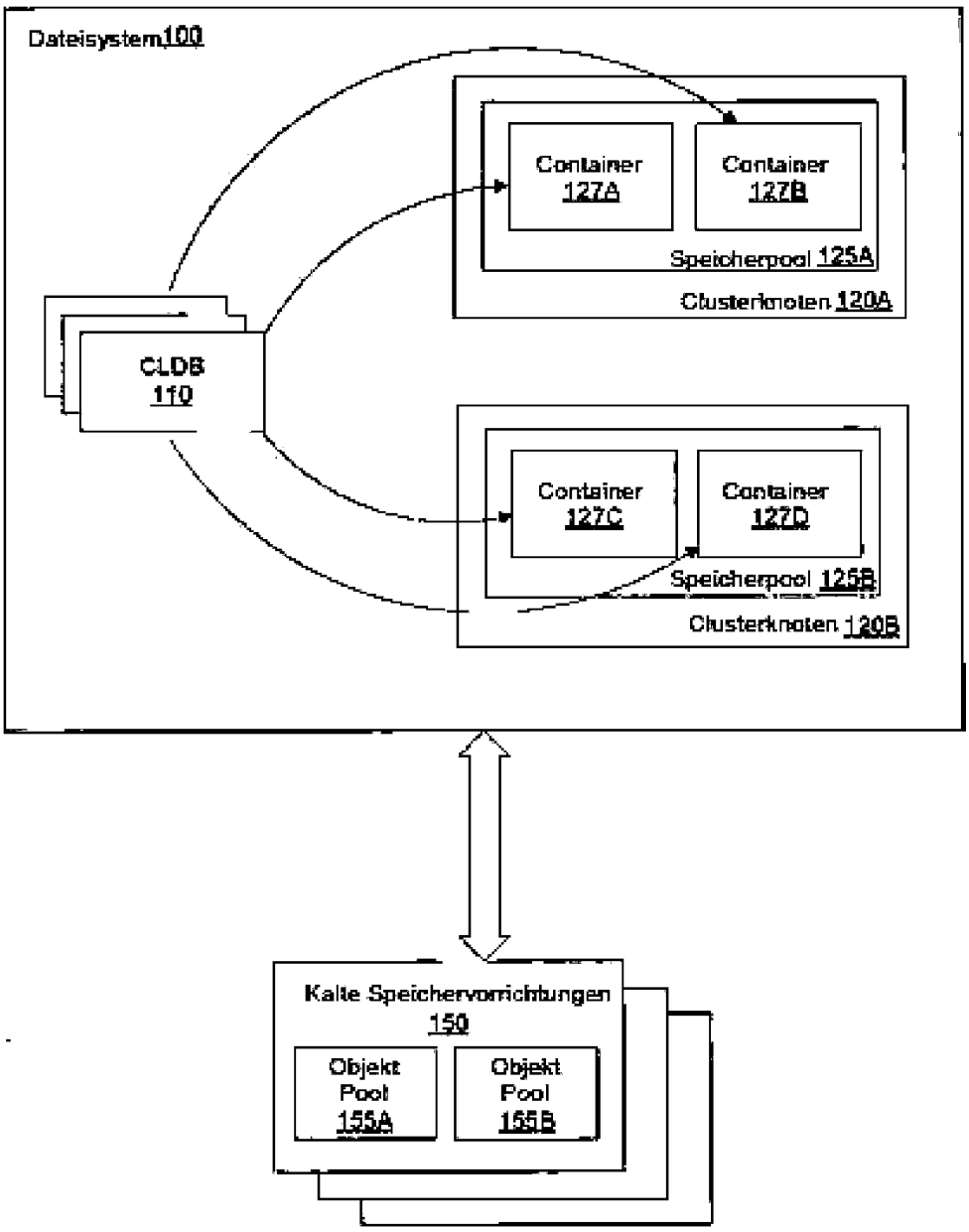


FIG. 1A

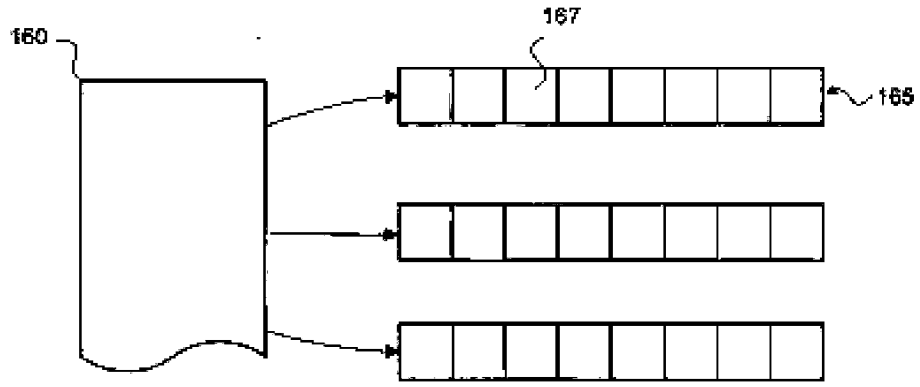

160
167
165

FIG. 1B

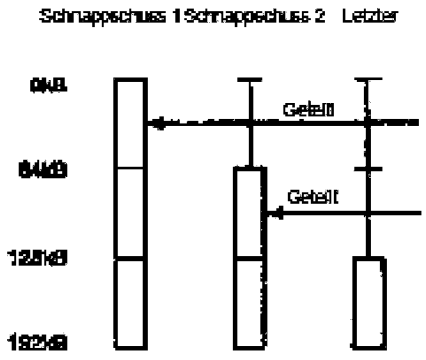

Schnappschuss 1
Schnappschuss 2
Letzter
0kB
64kB
128kB
192kB
Geteilt
Geteilt

FIG. 2A

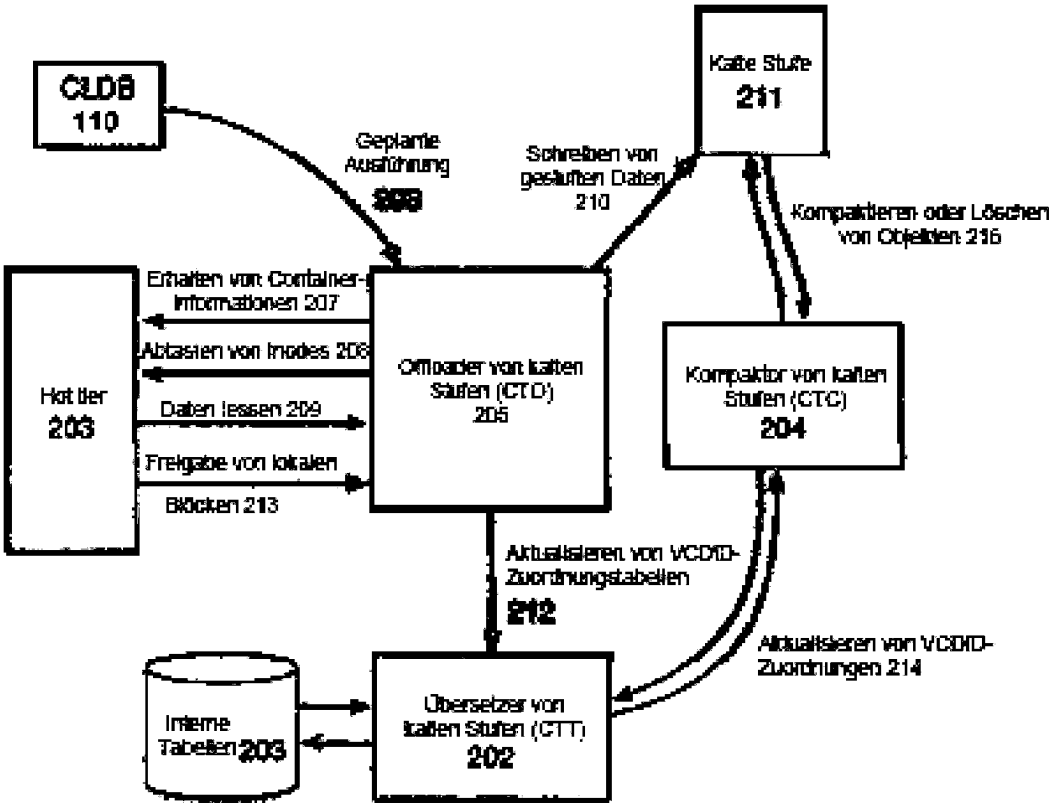

CLDB
110
Geplante
Ausführung
Kalte Stufe
211
Schreiben von
gesluften Daten
210
Kompaktieren oder Löschen
von Objekten 216
Erhalten von Container-
Informationen 207
Abtasten von Inodes 208
Daten lesen 209
Freigabe von lokalen
Blöcken 213
Heiß tier
203
Offloader von kalten
Stufen (CTO)
205
Kompaktor von kalten
Stufen (CTC)
204
Aktualisieren von VCDID-
Zuordnungstabellen
212
Aktualisieren von VCDID-
Zuordnungen 214
Interne
Tabellen 203
Übersetzer von
kalten Stufen (CTT)
202

FIG. 2B

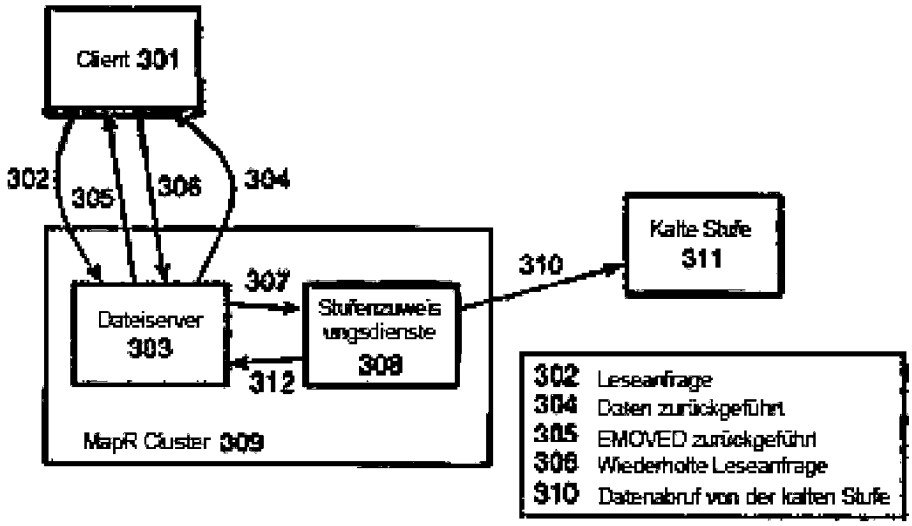


FIGURE. 3

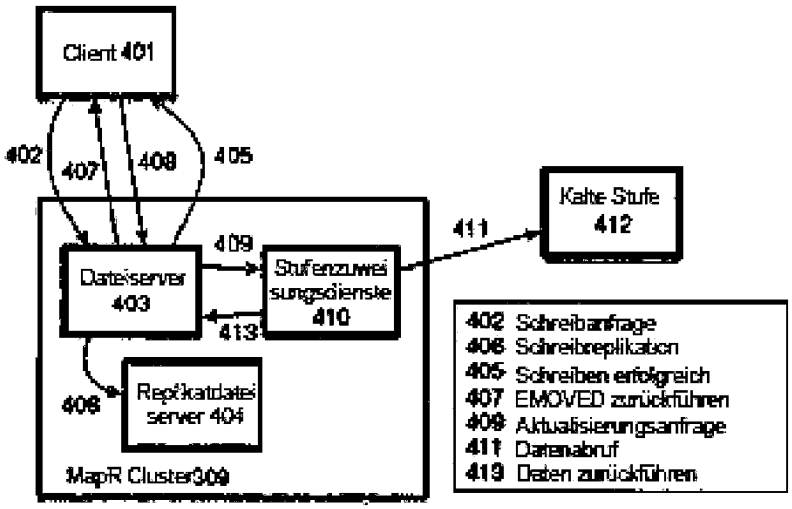


FIGURE. 4

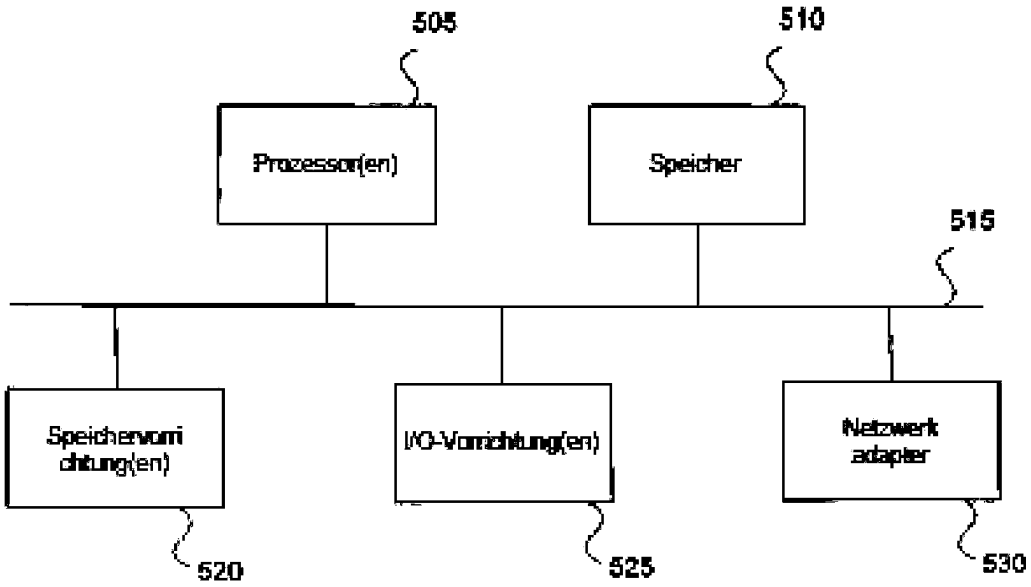

505
510
Prozessor(en)
Speicher
515
Speichervorrichtung(en)
I/O-Vorrichtung(en)
Netzwerkadapter
520
525
530

FIG. 5