(19) Deutsches Patent- und Markenamt

(10) **DE 11 2011 103 026 B4** 2022.03.03

(12) # Patentschrift

(21) Deutsches Aktenzeichen: **11 2011 103 026.6**
(86) PCT-Aktenzeichen: **PCT/US2011/039209**
(87) PCT-Veröffentlichungs-Nr.: **WO 2012/033554**
(86) PCT-Anmeldetag: **06.06.2011**
(87) PCT-Veröffentlichungstag: **15.03.2012**
(43) Veröffentlichungstag der PCT Anmeldung in deutscher Übersetzung: **13.06.2013**
(45) Veröffentlichungstag der Patenterteilung: **03.03.2022**

(51) Int Cl.: ***G06F 15/16*** (2006.01)
G06F 9/44 (2018.01)
G06F 9/4401 (2018.01)

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:
12/879,594 **10.09.2010** **US**

(73) Patentinhaber:
International Business Machines Corporation, Armonk, N.Y., US

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

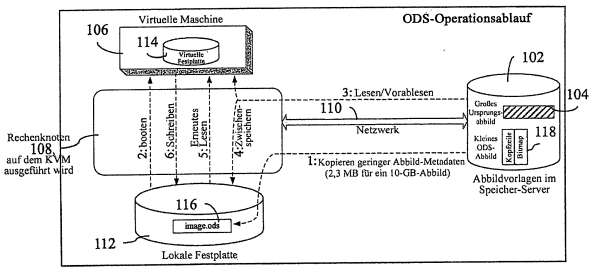
(72) Erfinder:
Tang, Chunqiang, Hawthorne, N.Y., US

(56) Ermittelter Stand der Technik:

US	**2004 / 0 034 671**	**A1**
US	**2011 / 0 083 006**	**A1**
US	**2011 / 0 107 044**	**A1**
US	**2011 / 0 246 985**	**A1**

(54) Bezeichnung: **Bedarfsgesteuertes Streaming von Abbildern virtueller Maschinen**

(57) Zusammenfassung: Ein bedarfsgesteuertes Abbild-Streaming (ODS) kann in einem Aspekt sowohl einen Vorgang des Erstellens einer Kopie beim Schreiben als auch einen Vorgang des Erstellens einer Kopie beim Lesen durchführen, um Daten auf einem Remote-Speicher-Server schrittweise auf eine lokale Festplatte eines Host zu übertragen. Ein Vorablesezugriff kann durchgeführt werden, während die Ressourcen ansonsten im Leerlauf sind, um Daten vom Remote-Speicher-Server auf die lokale Festplatte des Host zu übertragen. Ein neues Abbildformat und der entsprechende Blockeinheitentreiber für einen Hypervisor oder dergleichen können ebenfalls bereitgestellt werden. Das Abbildformat des ODS kann eine Kopfzeile und eine Bitmap beinhalten, die anzeigt, ob sich die Datensektoren auf einer lokalen Festplatte oder einem Remote-Speicher-Server befinden, und einen Abbild-Inhalt, der beispielsweise im Ursprungsformat gespeichert ist.



Beschreibung

GEBIET

[0001] Die vorliegende Anmeldung bezieht sich im Allgemeinen auf Computersysteme und insbesondere auf bedarfsgesteuertes Streaming von Abbildern virtueller Maschinen, beispielsweise für eine Cloud-Umgebung oder dergleichen.

HINTERGRUND

[0002] Bei einer Cloud-Computing-Umgebung kann der von einer virtuellen Maschine (VM) benötigte Blockeinheitenspeicher aus mehreren Quellen zugewiesen werden: Direktzugriffsspeicher (DAS, d.h. lokale Festplatte) des Host, Netzzugriffsspeicher (NAS, z.B. NFS) oder Speichernetzwerk (Storage Area Network (SAN)). Diese Optionen sind mit unterschiedlicher Leistung, Zuverlässigkeit und Verfügbarkeit zu unterschiedlichen Kosten verbunden.

[0003] Bei einem derzeit bekannten VM-Erstellungsverfahren wird die gesamte VM-Datei im Ursprungsformat (raw format, eine Byte-für-Byte-Kopie des Inhalts der physischen Blockeinheit) von einer in einem NAS gespeicherten Nur-Lese-Abbildvorlage in einen DAS eines Host kopiert. Nur dann könnte die VM des Host gebootet und ausgeführt werden. Diese Methodik ist mit einem Zeitaufwand für das Kopieren der gesamten Abbildvorlage in den DAS und somit mit einer langen Verzögerung verbunden, bis die neue VM gestartet und verwendet werden kann.

[0004] Bei einem weiteren bekannten Verfahren wird im DAS des Host lediglich ein Vorgang der Kopie beim Schreiben (Copy-on-Write) durchgeführt, d.h., nur geänderte Dateien werden im DAS gespeichert, nichtgeänderte Dateien hingegen werden stets aus dem Sicherungsabbild gelesen. Beim Verwenden einer im NAS als Sicherungsabbild gespeicherten Abbildvorlage kann eine VM womöglich schneller erstellt werden, da die Abbildvorlage beim Erstellen einer neuen VM nicht aus dem NAS in den DAS kopiert werden muss. Allerdings kann das wiederholte Lesen von nichtgeänderten Daten aus dem NAS übermäßigen Netzwerkdatenverkehr und E/A-Last im Share-NAS-Server erzeugen. Dies ist insbesondere bei einer Cloud-Umgebung mit mehreren VMs der Fall. Bei einem solchen Verfahren kann es erforderlich sein, dass die Cloud-Umgebung das Netzwerk und den NAS-Server mit jeweils ausreichender Kapazität für das Handhaben einer solchen Datenverkehrsmenge und E/A-Last bereitstellt.

[0005] Als weiteren Gesichtspunkt können die bestehenden Hypervisoren eine VM nur dann migrieren, wenn deren Abbilddatei in einem NAS gespeichert ist. Womöglich weil eine auf einem DAS laufende VM nicht migriert werden kann, kann ein Cloud-Anbieter den Benutzer einfach über einen anstehenden Wartungsvorgang auf einem Host benachrichtigen und auffordern, die Folgen eines VM-Verlusts zu akzeptieren. Das kann für den Cloud-Diensteanbieter zwar einfach sein, für Cloud-Benutzer unter Umständen aber nicht wünschenswert.

[0006] Die US 2011 / 0 246 985 A1 betrifft Systeme, Verfahren, Vorrichtungen und Computerprogrammprodukte, die alternative Desktop-Computing-Lösungen bieten und im Allgemeinen Client-Geräte bereitstellen, die so konfiguriert sind, dass sie entweder ein lokales gemeinsames Basisbild oder ein entfernt gespeichertes gemeinsames Basisbild verwenden, wobei ein benutzerspezifisches Overlay-Bild entfernt benutzerspezifische Daten speichert. Die Clients können so konfiguriert werden, dass sie das gemeinsame Basisbild lokal speichern.

[0007] Die US 2011 / 0 107 044 A1 betrifft ein Quellsystem, das einen Speicher und einen Prozessor umfasst, der einen Code ausführt, um zu veranlassen, dass zumindest ein Teil des Speichers über ein Netzwerk zu einem Zielsystem migriert wird. Der Prozessor veranlasst die Migration des Speichers, indem er einen Teil des Speichers migriert, während ein Gast weiterhin in den Speicher schreibt, die Ausführung des Gastes anhält und den Rest der Speichermigration abschließt.

[0008] Die US 2011 / 0 083 006 A1 betrifft einen Client-Terminal in einem Netzwerk-Boot-System. Der Client-Terminal wird dazu veranlasst, eine Revision einer virtuellen Festplatte zu erkennen, um Operationen eines Lese-Cache-Treibers zu entscheiden. Das Client-Terminal verfügt über ein physisches Speichergerät. Ein Betriebssystem, das auf dem Client-Terminal läuft, verfügt über einen Lese-Cache-Mechanismus, der Daten, die von einem Boot-Server ausgelesen wurden, im Lese-Cache speichert. Der Lese-Cache-Mechanismus umfasst einen Filtertreiber zur Umwandlung eines Zugriffs auf einen lokalen Bus des Client-Terminals in

einen Zugriff auf das Netzwerk und einen Lese-Cache-Treiber zur Ansteuerung eines Speichergeräts. Die Daten auf der virtuellen Festplatte werden mit der Revision der Lesecache-Daten verglichen. Die Operationen des Lese-Cache-Treibers werden auf der Grundlage des Vergleichsergebnisses entschieden.

[0009] Die US 2004 / 0 034 671 A1 betrifft eine zentralisierte Datenverwaltung für Personal Computer, die das selektive Hochfahren von einer entfernten Laufwerkskomponente umfasst. Nachfolgende Kopiervorgänge von Datenblöcken von der entfernten Laufwerkskomponente auf ein lokales Laufwerk in einem PC können durchgeführt werden, um ein aktualisiertes Festplattenabbild von der entfernten Festplatte zu erhalten. Nachfolgende Kopiervorgänge von Datenblöcken von dem lokalen Laufwerk auf die entfernte Festplatte können durchgeführt werden, um sicherzustellen, dass ein aktuelles Abbild der lokalen Festplatte auf der entfernten Festplatte erhalten bleibt.

KURZE ZUSAMMENFASSUNG

[0010] Die Erfindung betrifft Verfahren, ein System und ein nichtflüchtiges computerlesbares Speichermedium, deren Merkmalen in den entsprechenden unabhängigen Ansprüchen angegeben sind. Ausführungsformen der Erfindung sind in den abhängigen Patentansprüchen angegeben.

[0011] In einem Beispiel kann das Verfahren das Kopieren von mit einer ausgewählten virtuellen Maschine verknüpften Abbild-Metadaten von einem Speicher-Server, auf dem eine oder mehrere Abbildvorlagen (auch als Sicherungsabbilder bezeichnet) gespeichert sind, die jeweils einer oder mehreren virtuellen Maschinen entsprechen, in einen lokalen Speicher eines Host-Computers beinhalten, wobei der lokale Speicher des Host-Computers zunächst kein Abbild der ausgewählten virtuellen Maschine enthält. Das Verfahren kann auch das Booten der ausgewählten virtuellen Maschine im Host-Computer unter Verwendung der kopierten Abbild-Metadaten beinhalten, wodurch es der ausgewählten virtuellen Maschine ermöglicht wird, Daten aus der Abbildvorlage auf dem Speicher-Server zu lesen, die erforderlich sind, um das Ausführen der ausgewählten virtuellen Maschine auf dem Host-Computer fortzusetzen, wenn die benötigten Daten nicht im lokalen Speicher des Host-Computers gespeichert sind. Das Verfahren kann darüber hinaus das Kopieren der gelesenen Daten der Abbildvorlage vom Speicher-Server in den lokalen Speicher des Host-Computers beinhalten, wenn die gelesenen Daten der Abbildvorlage nicht in dem lokalen Speicher des Host-Computers gespeichert sind. Nachfolgende Lesevorgänge der gleichen Daten werden aus dem lokalen Speicher des Host--Computers durchgeführt. Das Verfahren kann auch das Setzen eines Bit in einer Bitmap beinhalten, um anzuzeigen, dass die gelesenen Daten im lokalen Speicher des Host-Computers gespeichert sind. Das Verfahren kann darüber hinaus noch das Nutzen einer Ressourcen-Leerlaufzeit beinhalten, um Daten der mit der ausgewählten virtuellen Maschine verknüpften Abbildvorlage (Sicherungsabbild) aus dem Speicher-Server in den lokalen Speicher des Host-Computers bereitzustellen (sog. „Prefetching“).

[0012] Ein Verfahren zum bedarfsgesteuerten Streaming von Abbildern virtueller Maschinen kann in einem weiteren Beispiel das Kopieren von mit einer virtuellen Maschine verknüpften Abbild-Metadaten von einem Quell-Computer, auf dem ein der virtuellen Maschine entsprechendes Abbild gespeichert ist, auf einen Ziel--Computer beinhalten, wobei der Ziel-Computer das Abbild der virtuellen Maschine zunächst nicht enthält. Das Verfahren kann darüber hinaus das Booten der virtuellen Maschine im Ziel-Computer unter Verwendung der kopierten Abbild-Metadaten und das Ermöglichen beinhalten, dass die virtuelle Maschine im Ziel-Computer Daten des Abbilds im Quell-Computer liest, die erforderlich sind, um das Ausführen der virtuellen Maschine im Ziel-Computer fortzusetzen, wenn die benötigten Daten des Abbilds nicht im Ziel-Computer gespeichert sind. Das Verfahren kann darüber hinaus das Kopieren der gelesenen Daten des Abbilds vom Quell-Computer auf den Ziel-Computer beinhalten, wenn die gelesenen Daten des Abbildes nicht im Ziel--Computer gespeichert sind, wobei bei nachfolgenden Lesevorgängen der gleichen Daten die kopierten Daten im Ziel-Computer gelesen werden. Das Verfahren kann ferner das Setzen eines Bit in einer Bitmap beinhalten, um anzuzeigen, dass die gelesenen Daten im Ziel-Computer gespeichert sind.

[0013] Ein System für das bedarfsgesteuerte Streaming von Abbildern virtueller Maschinen kann in einem Beispiel einen Ziel-Computer beinhalten, der zum Kopieren von mit einer virtuellen Maschine verknüpften Abbild-Metadaten von einem Quell-Computer funktionsfähig ist, auf dem eine der virtuellen Maschine entsprechende Abbildvorlage gespeichert ist, wobei der Ziel-Computer die Abbildvorlage der virtuellen Maschine zunächst nicht enthält und eine Speichereinheit, die lokal an den Ziel-Computer angeschlossen ist. Der Ziel--Computer kann darüber hinaus funktionsfähig sein, um die virtuelle Maschine im Ziel-Computer unter Verwendung der kopierten Abbild-Metadaten zu booten und es der virtuellen Maschine im Ziel-Computer zu ermöglichen, Daten der Abbildvorlage im Quell-Computer zu lesen, die erforderlich sind, um das Ausführen der virtuellen Maschine im Ziel-Computer fortzusetzen, wenn die benötigten Daten der Abbildvorlage nicht

im Ziel-Computer gespeichert sind. Der Ziel-Computer kann darüber hinaus funktionsfähig sein, um die gelesenen Daten der Abbildvorlage aus dem Quell-Computer in die lokal im Ziel-Computer angeschlossene Speichereinheit zu kopieren, wenn die gelesenen Daten der Abbildvorlage nicht im Ziel-Computer gespeichert sind, wobei nachfolgende Lesevorgänge der gleichen Daten aus der lokal im Ziel-Computer angeschlossenen Speichereinheit durchgeführt werden. Der Ziel-Computer kann darüber hinaus zum Setzen eines Bit in einer Bitmap funktionsfähig sein, um anzuzeigen, dass die gelesenen Daten im Ziel-Computer gespeichert sind.

[0014] Im Folgenden sind weitere Merkmale sowie die Struktur und der Betrieb verschiedener Ausführungsformen unter Bezugnahme auf die beiliegenden Zeichnungen ausführlich beschrieben. In den Zeichnungen bezeichnen gleiche Bezugszeichen identische oder funktional ähnliche Elemente.

Figurenliste

Fig. 1 ist ein Schaubild, das Systemkomponenten und einen Betriebsablauf zeigt, der beim bedarfsgesteuerten Streaming von virtuellen Maschinen (VM) der vorliegenden Offenbarung gemäß einer Ausführungsform durchgeführt wird.

Fig. 2 zeigt eine vereinfachte Ansicht eines ODS-Abbildformats gemäß einer Ausführungsform der vorliegenden Offenbarung.

Fig. 3 ist einen ODS-Treiber in einem KVM/QEMU-Stack gemäß einer Ausführungsform der vorliegenden Offenbarung.

Die **Fig. 4A** und **Fig. 4B** zeigen einen Vergleich einer VM-Erstellung unter Verwendung eines Ursprungsabbilds und eines ODS-Abbilds gemäß einer Ausführungsform der vorliegenden Offenbarung.

Fig. 5 zeigt Systemkomponenten und einen Betriebsablauf für eine Live-Migration einer virtuellen Maschine mit einem lokalen Speicher durch das ODS der vorliegenden Offenbarung.

Fig. 6 ist ein Ablaufplan, der ein Verfahren gemäß einer Ausführungsform zum Erstellen einer neuen VM zeigt.

Fig. 7 ist ein Ablaufplan, der ein Verfahren gemäß einer Ausführungsform für die Live-Migration einer neuen VM zeigt.

Die **Fig. 8A** bis **Fig. 8D** zeigen mögliche unterschiedliche Anwendungsfälle unterschiedlicher Merkmale des ODS der vorliegenden Offenbarung gemäß einer Ausführungsform.

AUSFÜHRLICHE BESCHREIBUNG

[0015] Ein bedarfsgesteuertes Abbild-Streaming (ODS) für Abbilder virtueller Maschinen kann bei einer Ausführungsform der vorliegenden Offenbarung einen Vorgang des Erstellens einer Kopie beim Schreiben (CoW), einen Vorgang des Erstellens einer Kopie beim Lesen (CoR) und einen Vorablesezugriff durchführen. Der Vorgang des Erstellens einer Kopie beim Schreiben vermeidet, dass ein Datensektor wiederholt aus einem Remote-Speicher-Server (z.B. Netzzugriffsspeicher, NAS) gelesen wird, indem eine Kopie des zurückgegebenen Sektors auf der lokalen Festplatte (z.B. Direktzugriffsspeicher, DAS) der Computer-Host-Maschine zur Verwendung gespeichert wird. Beim Vorablesezugriff wird eine Leerlaufzeit genutzt, um den Rest des Abbilds, auf den von der virtuellen Maschine nicht zugegriffen wurde, von einem Remote-Speicher-Server (z.B. NAS) auf die lokale Festplatte (z.B. DAS) zu kopieren. Während der Vorgänge des Erstellens einer Kopie beim Schreiben und des Erstellens einer Kopie beim Lesen stellt die vorliegende Offenbarung bei einer Ausführungsform das Aktualisieren sowohl von Daten als auch Metadaten auf einer Festplatte bereit, wobei die Metadaten anzeigen, dass die Daten jetzt auf der lokalen Festplatte (z.B. DAS) und nicht auf dem Remote-Speicher-Server (z.B. NAS) gespeichert sind.

[0016] Das ODS der vorliegenden Offenbarung kann bei einer Ausführungsform ein neues Abbildformat und den entsprechenden Blockeinheitentreiber für QEMU beinhalten. Das ODS der vorliegenden Offenbarung kann bei einer Ausführungsform für virtuelle Maschinen konzipiert sein, deren Abbilder im Direktzugriffsspeicher des Host-Computers gespeichert sind. Die Hauptanwendungsfälle des ODS können (1) das sofortige Erstellen einer virtuellen Maschine (VM) im Direktzugriffsspeicher (z.B. DAS, d.h. lokale Festplatte des Host), ohne dass auf den Abschluss des Kopiervorgangs der Abbildvorlage der VM vom Remote-Speicher--Server in den DAS gewartet wird, und (2) die Live-VM-Migration zwischen Maschinen, die DAS verwenden, um VMs zu betreiben, beinhalten.

[0017] Das ODS der vorliegenden Offenbarung kann gemäß einer Ausführungsform sowohl einen Vorgang des Erstellens einer Kopie beim Schreiben als auch einen Vorgang des Erstellens einer Kopien beim Lesen durchführen, um Daten schrittweise vom Remote-Speicher-Server auf die lokale Festplatte eines Host zu übertragen. Bei einer Cloud-Umgebung mit einer großen Anzahl von VMs vermeidet der Vorgang des Erstellens einer Kopie beim Lesen, dass der gleiche Datensektor wiederholt aus dem Remote-Speicher-Server gelesen wird, wodurch übermäßiger Netzwerkdatenverkehr oder Eingabe-/Ausgabe-(E/A-)Last im Speicher--Server erzeugt werden kann. Das ODS der vorliegenden Offenbarung kann gemäß einer Ausführungsform auch einen Vorablesezugriff durchführen. Es identifiziert Leerlaufzeit, um noch unberührte Abbilddaten vom Remote-Speicher-Server auf die lokale Festplatte des Host zu kopieren. Das Abbildformat des ODS kann bei einer Ausführungsform (1) eine Kopfzeile, (2) eine Bitmap, die anzeigt, ob die Datensektoren auf einer lokalen Festplatte oder im Remote-Speicher-Server liegen, und (3) den im Ursprungsformat gespeicherten Abbildinhalt beinhalten.

[0018] **Fig. 1** ist ein Schaubild, das Systemkomponenten und einen Betriebsablauf zeigt, der vom bedarfsgesteuerten Streaming von virtuellen Maschinen (VM) der vorliegenden Offenbarung gemäß einer Ausführungsform durchgeführt wird. Ein Speicher-Server 102 speichert eine oder mehrere Abbildvorlagen 104 einer oder mehrerer entsprechender virtueller Maschinen. Der Speicher-Server 102 ist in Bezug auf einen Host-Computer 108, der eine virtuelle Maschine 106 betreibt und über ein Netzwerk 110 verbunden ist, entfernt angeordnet. Ein Beispiel für den Speicher-Server 102 kann ein NAS-Server sein. Der Speicher-Server 102 kann gemeinsam genutzt werden (z.B. ein gemeinsam genutzter Speicher-Server, SONAS) und kann über ein Network File System (NFS) exportierte Abbildvorlagen beinhalten. Ein Beispiel für eine lokale Festplatte 112 kann DAS sein. Ein DAS kann als flüchtiger Speicher angesehen werden, wogegen der NAS als flüchtiger Speicher erachtet werden kann. Ein flüchtiger Speicher kann verwendet werden, um ein Festplattenabbild einer VM, das das Stammdateisystem enthält, zu speichern. Wenn die VM erlischt, gehen Daten in einem flüchtigen Speicher verloren. Optional kann ein Benutzer einen permanenten Speicher an eine VM anschließen. Ein permanenter Speicher besteht über die Lebensdauer einer VM hinaus und kann beispielsweise verwendet werden, um permanente Daten einer zugehörigen Datenbank zu speichern. Bei einer Cloud-Umgebung wird ein flüchtiger Speicher von den lokalen Festplatten eines Host bereitgestellt, möglicherweise ohne Hardware-RAID.

[0019] VMs können auf der Grundlage von Nur-Lese-Abbildvorlagen 104 erstellt werden, die auf einem Speicher-Server 102 gespeichert und für alle Hosts (z.B. Computer oder Maschine, der bzw. die VMs betreibt oder auf dem bzw. der VMs ausgeführt werden) zugänglich sind. Eine virtuelle Festplatte 114 einer VM kann als reguläre Datei 116 im Dateisystem des Host gespeichert werden. Ein Host-Computer (ein Rechenknoten) 108 kann einen Hypervisor wie KVM beinhalten. Ein Hypervisor (oder ein VM-Monitor) ist eine Software-Komponente, die ermöglicht, dass mehrere Betriebssysteme gleichzeitig auf einer bestimmten Maschine (Hardware oder Prozessor) ausgeführt werden können. Je nach Hypervisor können mehrere Formate für virtuelle Festplatten 114 unterstützt werden. Beispielsweise unterstützt KVM/QEMU mehrere Formate für virtuelle Festplatten. KVM ist eine Linux-Kernel-Virtualisierungsinfrastruktur. Sie nutzt QEMU für die E/A-Emulation. Ein Ursprungsformat ist eine Byte-für-Byte-Kopie des in einer regulären Datei gespeicherten Inhalts einer physischen Blockeinheit. QCOW2 ist ein weiteres Abbildformat, das vom QEMU unterstützt wird. Das QCOW2-Abbild speichert nur geänderte Daten, nichtgeänderte Daten werden hingegen stets aus dem Sicherungsabbild (d.h. Speicher-Server, z.B. NAS) gelesen.

[0020] Zunächst enthält die lokale Festplatte 112 des Host-Computers 108 keine Abbildvorlage zum Ausführen einer ausgewählten virtuellen Maschine 106. Das ODS der vorliegenden Offenbarung kopiert gemäß einer Ausführungsform in Reaktion auf den Empfang einer Anweisung zum Starten oder Booten einer VM 106 kleine Abbild-Metadaten 118 vom Speicher-Server 104 auf die lokale Festplatte 112, wie bei 116 gezeigt. Die Abbild-Metadaten beinhalten eine Kopfzeile und eine Bitmap. Die Kopfzeile identifiziert die Abbildvorlage, und die Bitmap wird verwendet, um jene Abschnitte (z.B. Sektoren) der Abbildvorlage zu identifizieren, die lokal gespeichert sind. Im Ausgangszustand identifiziert die Bitmap gemäß einer Ausführungsform die Sektoren der Abbildvorlage, die zur Gänze mit Nullen gefüllt sind. Zur Laufzeit besteht kein Bedarf, diese mit Nullen gefüllten Sektoren vom Speicher-Server 104 auf der lokalen Festplatte 112 zu kopieren. Das ODS der vorliegenden Offenbarung kann bei einer weiteren Ausführungsform den Schritt des Kopierens von kleinen Abbild-Metadaten 118 aus dem Speicher-Server 104 auf die lokale Festplatte 112 auslassen, wie bei 116 gezeigt. In diesem Fall werden die Metadaten auf der lokale Festplatte 112 völlig neu erstellt, wobei alle Bits in der Bitmap so gesetzt werden, dass sie anzeigen, dass kein Datensektor lokal gespeichert ist. Die formale Ausführungsform hat den Vorteil, dass keine mit Nullen gefüllten Sektoren in die Abbildvorlage kopiert werden. Die VM 106 wird unter Verwendung der Abbild-Metadaten 118 gebootet, und wenn die VM 106 auf zusätzliche Daten 104 zugreift und diese aus dem Speicher-Server 102 liest, um sich zu booten und auszu-

führen, werden diese Daten als lokales Abbild 116 ebenfalls auf der lokale Festplatte 112 kopiert oder dort gespeichert. Das eine oder die mehreren Bits in der Bitmap werden ebenfalls aktualisiert, so dass sie anzeigen, dass die entsprechenden Datenabschnitte oder -sektoren der Abbildvorlage 104 lokal gespeichert wurden. Das nächste Mal, wenn die VM 106 auf die gleichen Daten zugreifen muss, wird die lokal gespeicherte Version verwendet und nicht über ein Netzwerk auf die Abbildvorlage 104 im Speicher-Server 102 zugegriffen.

[0021] Der Laufzeitbetrieb des ODS der vorliegenden Offenbarung kann die asynchronen Vorgänge des Erstellens einer Kopie beim Lesen, des Erstellens einer Kopie beim Schreiben und Vorablesezugriffe an Abbilddaten über das Netzwerk im Hintergrund beinhalten. Bei einem asynchronen Erstellen einer Kopie beim Lesen ruft ein um das ODS der vorliegenden Offenbarung erweiterter Hypervisor auf der Host-Maschine 108, wenn die VM 106 einen Sektor das erste Mal liest, den Sektor über das Netzwerk 110 aus dem Remote-Speicher-Server 102 ab. Im Hintergrund speichert der um das ODS der vorliegenden Offenbarung erweiterte Hypervisor im Host-Computer 108 bei einer Ausführungsform den Sektor in seiner lokalen ODS-Datei 116 und setzt die Bitmap entsprechend. Die Bitmap ist Teil des ODS-Abbilds und wird auf der lokalen Festplatte gespeichert. Bei nachfolgenden Lesevorgängen des Sektors werden die Daten bei einer Ausführungsform der vorliegenden Offenbarung stets direkt aus der lokalen ODS-Datei 116 bezogen. Um direkt auf die lokale Festplatte zu schreiben (Erstellen einer Kopie beim Schreiben), wenn die VM einen Sektor schreibt, schreibt der um das ODS der vorliegenden Offenbarung erweiterte Hypervisor im Host-Computer 108 gemäß einer Ausführungsform direkt in die lokale ODS-Datei 116, ohne dass 4 KB Daten aus dem Speicher--Server 102 abgerufen werden müssen. Das ODS der vorliegenden Offenbarung kann gemäß einer Ausführungsform auch einen Vorablesezugriff von Abbilddaten über das Netzwerk im Hintergrund durchführen. Im Rahmen einer konfigurierbaren Richtlinie kann der um das ODS der vorliegenden Offenbarung erweiterte Hypervisor im Host-Computer 108 als Hintergrundoperation gemäß einer Ausführungsform die gesamten Ursprungsabbilddaten 104 über das Netzwerk 110 mittels Streaming vorab lesen und die Daten in der lokalen ODS-Datei 116 speichern. Beispielsweise kann eine Vorablesezugriff-Richtlinie vorgeben, dass der Vorablesezugriff beginnen kann, nachdem die VM 106 für 12 Stunden ausgeführt wurde, sowie um Mitternacht, wenn die Arbeitslast im Netzwerk 110 und im Speicher-Server 102 gering ist. Folglich werden gemäß dieser beispielhaften Richtlinie keine Daten für eine VM vorab gelesen, deren Lebensdauer unter 12 Stunden liegt. Die Richtlinie kann gemäß einer Ausführungsform von einem Systemadministrator oder anderen Benutzern festgelegt werden.

[0022] **Fig. 2** zeigt eine vereinfachte Ansicht eines ODS-Abbildformats gemäß einer Ausführungsform der vorliegenden Offenbarung. Eine virtuelle Festplatte im ODS-Format kann in **Fig. 2** jeder VM zugeordnet werden, die auf einer Host-Maschine oder einem Host-Computer ausgeführt wird. In einem Aspekt basiert das ODS-Abbild auf einem Nur-Lese-Sicherungsabbild. Die ODS-Kopfzeile 202 speichert einen Verweis zum Sicherungsabbild 206, um die Abbildvorlage der virtuellen Maschine zu identifizieren. Ein Sicherungsabbild bezieht sich auf eine im permanenten Speicher gespeicherte Abbildvorlage einer virtuellen Maschine. Der Verweis kann der Dateiname der Abbildvorlage der virtuellen Maschine sein, z.B. eine Zeichenfolge, die den Namen des Ursprungsabbilds speichert, auf dem das ODS-Abbild basiert. Andere Verweise können verwendet werden, um die Abbildvorlage der virtuellen Maschine zu identifizieren. Das ODS-Abbildformat kann darüber hinaus eine Bitmap 204 beinhalten, beispielsweise mit einem Bit für jeden Datensektor im Abbild der virtuellen Festplatte. Das Bit zeigt an, ob der aktuelle Inhalt des entsprechenden Sektors im ODS-Abbild oder im Sicherungsabbild gespeichert ist. Beispielsweise wird ein Bit auf 1 gesetzt, wenn sein entsprechender Sektor bereits vom Ursprungsabbild in das ODS-Abbild kopiert wurde, oder wenn der Sektor von der VM bereits lokal geschrieben wurde. Die Größe der Bitmap ist bei einer Ausführungsform der vorliegenden Offenbarung proportional zur Größe der Abbildvorlage. Ein ODS-Blockeinheitentreiber ist beispielsweise in QEMU umgesetzt, der das ODS-Format unterstützt und von der VM ausgegebene Festplatten-Eingabe-/Ausgabe--Anfragen (E/A-Anfragen) bearbeitet. Der Speicherplatz für die Abbilddaten 208 kann zunächst leer sein; zu Beginn liegen keine Daten vor und es muss kein Speicherplatz reserviert werden. Die Größe des Speicherplatzes für die Abbilddaten 208 kann genauso groß wie jener der Abbildvorlage werden, aus der die Abbilddaten kopiert werden.

[0023] Das ODS-Abbildformat kann darüber hinaus Speicherplatz für erweiterte Festplattendaten 210 beinhalten. Der Speicherplatz für erweiterte Festplattendaten 210 kann willkürlicher Größe sein, und die Größe kann jederzeit verändert werden, um eine Größenänderung des Abbilds zu unterstützen. In einem Vorlagen-Abbild sind für diesen Speicherplatz 210 keine entsprechenden Daten vorhanden. Ferner wird keine Bitmap benötigt, die den Daten in diesem Speicherplatz 210 entspricht. Somit hängt die Größe der Bitmap nicht von der Größe des Speicherplatzes für die erweiterten Festplattendaten 210 ab; eine Größenänderung des ODS-Abbilds bedingt durch den Speicherplatz für die erweiterten Festplattendaten 210 beeinflusst die Bitmap

nicht. Die Größenänderung eines ODS-Abbilds, um „Speicherplatz für erweiterte Festplattendaten“ zu schaffen, ist unabhängig von der Größe der „erweiterten Festplattendaten“ eine konstante Zeitoperation. Es muss lediglich das Feld „disk_size“ in der „Kopfzeile“ eines ODS-Abbilds verändert werden. Die Daten in diesem Speicherplatz 210 können nur lokal verwendet werden. Der Speicherplatz für erweiterte Festplattendaten ist optional.

[0024] Um eine neue VM zu starten, erstellt der Host auf seiner lokalen Festplatte (z.B. DAS) ein ODS-Abbild, dessen Verweis auf ein Sicherungsabbild auf die im Speicher-Server (z.B. NAS) gespeicherte Abbildvorlage 206 zeigt. Beispielsweise kann ein ODS-Abbild, das nur die Kopfzeile und die Bitmap enthält, auf die lokale Festplatte (z.B. DAS) kopiert werden. Die VM kann dann sofort gebootet werden, ohne dass Abbilddaten (Vorlagen-Abbild) vom Speicher-Server (z.B. NAS) auf die lokale Festplatte (z.B. DAS) kopiert werden, d.h., der Abschnitt 208 „Speicherplatz für Festplattendaten“ des ODS-Abbilds ist zu Beginn leer. Beispielsweise kann die Bitmap für ein Original-Ursprungsabbild von 10 Gigabyte (GB) nur 2,5 Megabyte (MB) groß sein. Nur die 2,5-MB-Bitmap plus die kleinen Kopfzeilenfelder müssen über das Netzwerk kopiert werden, wenn eine neue VM erstellt und gebootet wird. Bei Bearbeitung einer Festplatten-Schreibanfrage von der VM kann der ODS-Treiber des QEMU die Daten im ODS-Abbild 208 speichern und die Bitmap 204 entsprechend aktualisieren. Dieses Verhalten wird als „Erstellen einer Kopie beim Schreiben“ bezeichnet.

[0025] Beim Bearbeiten einer Festplatten-Leseanfrage von der VM überprüft der ODS-Treiber die Bitmap 204, um zu ermitteln, ob sich die angeforderten Daten im ODS-Abbild befinden. Ist dies der Fall, werden die Daten aus dem ODS-Abbild gelesen und an die VM zurückgegeben. Ist dies nicht der Fall, werden die Daten aus dem Sicherungsabbild 206 gelesen und an die VM zurückgegeben. Während die VM die zurückgegebenen Daten weiterhin im Hintergrund bearbeitet, wird eine Kopie der zurückgegebenen Daten im ODS-Abbild 208 gespeichert, und die Bitmap 204 wird entsprechend aktualisiert, so dass die Daten bei künftige Leseanfragen für die gleichen Daten aus dem ODS-Abbild von der lokalen Festplatte (z.B. DAS) und nicht aus dem Sicherungsabbild 208 im Speicher-Server (z.B. NAS) bezogen werden. Dieses Verhalten wird als „Erstellen einer Kopie beim Lesen“ bezeichnet. Bei diesem Verhalten des Erstellens einer Kopie beim Lesen kann ein Datensektor höchstens einmal aus dem Speicher-Server (z.B. NAS) gelesen werden, wodurch übermäßiger Netzwerkdatenverkehr und E/A-Last im Speicher-Server (z.B. NAS) besser vermieden werden können.

[0026] Beim Vorgang des Erstellens einer Kopie beim Lesen wird der Inhalt des Sicherungsabbilds 206 schrittweise vom Speicher-Server (z.B. NAS) auf die lokale Festplatte (z.B. DAS) des Host-Computers migriert. Der Vorablesemechanismus des ODS kann darüber hinaus Leerlaufressourcen nutzen, um die Datenmigration zu beschleunigen. Tabelle 1 zeigt ein Beispiel für das ausführliche Layout eines ODS-Abbilds auf einer Festplatte gemäß einer Ausführungsform der vorliegenden Offenbarung.

Tabelle 1

```
Struct OdsImage Layout {
uint32_t        magic;
uint32_t        version;
uint64_t        ods_header_size;
int64_t         disk_data_size;
char            backing_image[1024]
int64_t         effective_backing_image_size;
uint32_t        all_data_in_ods_image;
uint32_t        bitmap_unit;
uint32_t        page_size;
... /* einige andere Felder übersprungen */
char            separate_disk_data_file[1024]
uint8_t         padding_to_page_boundary_1 [..];
uint8_t         bitmap [...];
uint8_t         padding_to_page_boundary_2[..];
uint8_t         disk_data[...];
```

```
Struct OdsImage Layout {
}
```

[0027] Im Abschnitt „disk_data“ können Festplatteninhalte im Ursprungsformat gespeichert werden. Der Inhalt des Sektors mit der logischen Blockadresse LBA = n kann unter disk_data[n*512] gespeichert werden, wobei 512 die Sektorgröße ist.

[0028] **Fig. 3** ist ein ODS-Treiber in einem KVM/QEMU-Stack (ein beispielhafter Hypervisor) gemäß einer Ausführungsform der vorliegenden Offenbarung. Ein Betriebssystem-Kernel 302, z.B. ein Linux-Kernel, kann auf einem Hardware-Prozessor 304 ausgeführt werden. Der Betriebssystem-Kernel 302 kann eine Vielzahl von Hypervisor-Programmen ausführen, z.B. KVM, eine unterschiedliche Anbieterversion von KVM usw. Ein weiteres Beispiel für einen Hypervisor ist Xen. Ein Hypervisor-Programm 306 (z.B. KVM) ermöglicht, dass eine virtuelle Maschine 310 im Hardware-Prozessor 304 ausgeführt werden kann. Darüber hinaus kann ein Hypervisor-Programm 306 mehrere virtuelle Abbildformate 308 unterstützen, z.B. raw, qcow2 und das ODS der vorliegenden Offenbarung. Beim Booten einer virtuellen Maschine 310 versucht die virtuelle Maschine, den ersten Sektor (z.B. 512 Kilobyte (KB)) auf der entsprechenden virtuellen Festplatte zu lesen (die Abbildformate 308 auf einer lokalen Festplatte). Ein Hypervisor-Programm, das erweitert wurde, um mit der Funktion des ODS der vorliegenden Offenbarung zu funktionieren, erkennt, dass das ODS-Abbild noch nicht auf der virtuellen Festplatte (lokale Festplatte) vorhanden ist und beginnt mit der ODS-Methodik der vorliegenden Offenbarung (führt diese aus). Beim ODS der vorliegenden Offenbarung wird ein Streaming der erforderlichen Daten durchgeführt, wie hier beschrieben, wobei die VM gemäß einer Ausführungsform beinahe sofort gebootet wird. Für die Umsetzung kann ein Hypervisor-Benutzeradressbereichs-Programm (z.B. ein qemu-kvm-Benutzeradressbereichs-Programm) verändert werden, um ods.c und ods.h hinzuzufügen, ohne andere Codes dabei zu verändern. Insbesondere sind keine Änderungen im Kernel des Computerknotens (z.B. Linux) erforderlich.

[0029] Die **Fig. 4A** und **Fig. 4B** zeigen Vergleiche hinsichtlich der VM-Erstellung unter Verwendung eines Ursprungsabbilds und eines ODS-Abbilds. Im gezeigten Beispiel werden drei VMs gleichzeitig erstellt. Wie in **Fig. 4A** ersichtlich, muss beim VM-Erstellungsverfahren unter Verwendung des Ursprungsabbildformats gewartet werden, bis eine gesamte Abbildvorlage vom Speicher-Server (z.B. NAS) auf die lokale Festplatte (z.B. DAS) kopiert worden ist, und erst dann wird die VM gebootet. Dies ist sogar der Fall, wenn ein Großteil der kopierten Abbilddaten während des VM-Bootvorgangs nicht erforderlich ist und während der gesamten Lebensdauer der VM womöglich nicht einmal auf diesen zugegriffen wird. Das ODS-Erstellungsverfahren der vorliegenden Offenbarung gemäß einer Ausführungsform ist in **Fig. 4B** gezeigt. Das ODS bootet die VM sofort, ohne dass sich Abbilddaten auf der lokalen Festplatte (z.B. DAS) befinden, und kopiert Daten nach Bedarf, d.h., wenn die VM darauf zugreift, vom Speicher-Server (z.B. NAS) auf die lokale Festplatte (z.B. DAS) des Host. Darüber hinaus identifiziert der Vorablesezugriff-Mechanismus des ODS Ressourcen-Leerlaufzeit, um den Rest des Abbilds oder Abschnitte des Abbilds, auf den bzw. auf die die VM noch nicht zugegriffen hat, vom Speicher-Server (z.B. NAS) auf die lokale Festplatte (z.B. DAS) zu kopieren. Der Vorablesezugriff kann dahingehend konservativ sein, dass das ODS, wenn es einen Konflikt bei Ressourcen erkennt (z.B. Speicher-Server, lokale Festplatte oder Netzwerk), den Vorablesezugriff vorübergehend anhalten und später wiederaufnehmen kann, wenn sich der Stau aufgelöst oder verringert hat. Bei einer weiteren Ausführungsform kann die vorliegende Offenbarung, nachdem Abbildvorlagen mithilfe des Vorablesezugriffs zur Gänze auf die lokale Festplatte kopiert wurden, gängige Abbildvorlagen auf einer lokalen Festplatte (z.B. DAS) des Host zwischenspeichern, so dass die zwischengespeicherten Abbildvorlagen zur sofortigen Erstellung von neuen VMs verwendet werden können, d.h., der Abbild-Kopierschritt wird bei einem Treffer im Cachespeicher übersprungen.

[0030] Das ODS der vorliegenden Offenbarung führt gemäß einer Ausführungsform einen Vorgang des Erstellens einer Kopie beim Lesen aus, um Datensektoren nach Datenbedarf vom Speicher-Server (z.B. NAS) in einen lokalen Speicher (z.B. DAS) zu übertragen. Optional kann beim ODS ein Vorablesezugriff aktiviert werden, um noch unberührte Datensektoren mithilfe nichtgenutzter Ressourcen im Leerlauf vom Speicher-Server (z.B. **Fig. 1** bei 102) in den lokalen Speicher (z.B. **Fig. 1** bei 112) zu kopieren. Das ODS der vorliegenden Offenbarung kann gemäß einer Ausführungsform auch einen Vorablesezugriff des gesamten Abbilds durchführen. Beim Vorablesezugriff des gesamten Abbilds nutzt das ODS der vorliegenden Offenbarung gemäß einer Ausführungsform Leerlaufressourcen, um das gesamte Abbild sequentiell, beispielsweise vom ersten Datensektor zum letzten Datensektor, vom Speicher-Server (z.B. NAS) in einen lokalen Speicher (z.B. DAS) zu kopieren. Gemäß einer Ausführungsform können Datensektoren, auf die noch nicht zugegriffen wurde und die eng mit Datensektoren in Zusammenhang stehen, auf die bereits zugegriffen wurde, bei einem stand-

ortbasierten Vorablesezugriff optional vorab gelesen werden, anstatt das gesamte Abbild einem sequentiellen Vorablesezugriff zu unterziehen. Gemäß einer Ausführungsform können Datensektoren, auf die noch nicht zugegriffen wurde, bei einem profilbasierten Vorablesezugriff optional entsprechend einem offline erstellten Abbildvorlagenprofil vorab gelesen werden. Das Profil kann die Reihenfolge des Vorablesezugriffes für Datensektoren auf der Grundlage ihrer Wichtigkeit priorisieren oder Datensektoren identifizieren, auf die für gewöhnlich gemeinsam zugegriffen wird, d.h., es wird ein Arbeitssatz unter Verwendung der Caching-Terminologie erstellt. Wenn ein Datensektorstatus bereits S_{in_ods} ist, wird dieser Datensektor beim Vorablesezugriff übersprungen. Nach abgeschlossenem Vorablesezugriff setzt es eine Markierung, z.B. all_data_in_ods_image, im ODS-Abbild (siehe z.B. Tabelle 1). Dann können alle gelesenen und geschriebenen, von der VM ausgegebenen Anfragen direkt im disk_data-Abschnitt des ODS-Abbilds arbeiten, ohne dass die Bitmap überprüft oder aktualisiert werden muss, da von vornherein bekannt ist, dass Daten im Sicherungsabbild nicht mehr benötigt werden.

[0031] Da die Daten, die vorab gelesen werden, nicht dringend benötigt werden, wird im Rahmen des Vorablesezugriffs versucht, Konflikte mit Ressourcen wie Prozessoren (z.B. CPU), Festplatten (z.B. DAS, NAS) und Netzwerk zu vermeiden. Der ODS-Treiber überwacht die Reaktionszeit Trfür das Lesen von Daten aus einem Speicher-Server (z.B. NAS) sowie die Reaktionszeit Twfür das Schreiben von Daten auf eine lokale Festplatte (z.B. DAS). Wenn Tr < Cr und Tw < Cw, setzt der ODS-Treiber den Vorablesezugriff fort, wobei Cr und Cw zwei konstante Grenzwerte sind, z.B. Cr = 30 Millisekunden (ms) und Cw = 30 ms. Wenn eine Reaktionszeit über dem Grenzwert liegt, generiert er eine zufällige Zahl, um eine Entscheidung darüber zu treffen, ob der Vorablesezugriff angehalten werden soll. Bei einer Ausführungsform setzt er den Vorablesezugriff mit einer 50%igen Wahrscheinlichkeit fort und hält ihn mit einer 50%igen Wahrscheinlichkeit für einen fixen Zeitraum an. Wenn er sich für ein Anhalten des Vorablesezugriffs entscheidet, setzt er diesen später fort, um zu überprüfen, ob sich der Ressourcenkonflikt aufgelöst hat, indem er eine kleine Datenmenge versuchsweise von einem Speicher-Server (z.B. NAS) auf eine lokale Festplatte (z.B. DAS) kopiert. Er überwacht die Reaktionszeiten und entscheidet, ob der Vorablesezugriff fortgesetzt oder weiter pausiert wird.

[0032] Da die Entscheidung über ein Anhalten des Vorablesezugriffs willkürlich erfolgt, wenn die Reaktionszeit über dem Grenzwert liegt, halten bei einer Ausführungsform der vorliegenden Offenbarung 50 % der aktiv einen Vorablesezugriff ausführenden ODS-Instanzen, wenn mehrere ODS-Instanzen miteinander konkurrieren, den Vorablesezugriff nach jedem Durchlauf einer Vorablesezugriff-Operation an, bis entweder alle ODS-Instanzen den Vorablesezugriff einstellen oder der Engpass der Ressourcen aufgehoben ist.

[0033] Um Störungen zu verringern, werden die Reaktionszeiten als exponentielle gleitende Durchschnitte berechnet.

$$T_r^{(n+1)} = 0.9T_r^{(n)} + 0.1\,S_{r\_new\_sample} \qquad (1)$$

$$T_w^{(n+1)} = 0.9T_w^{(n)} + 0.1\,S_{w\_new\_sample} \qquad (2)$$

[0034] Im Gegensatz zu Ressourcenverwaltungssystemen, die mehrere gleichzeitige Festplatten-E/A-Anfragen senden, kann der Vorablesezugriff beim ODS der vorliegenden Offenbarung gemäß einer Ausführungsform in Bezug auf die Ressourcennutzung konservativ sein. Beispielsweise behält es höchstens eine ausstehende Leseanfrage an das Sicherungsabbild auf einem Speicher-Server (z.B. NAS) und höchstens eine ausstehende Schreibanfrage für das ODS-Abbild auf einer lokalen Festplatte (z.B. DAS). Nachdem es einige Datensektoren aus dem Speicher-Server (z.B. NAS) vorab gelesen hat, sendet es die nächste Leseanfrage sofort an den Speicher-Server (z.B. NAS) und schreibt gleichzeitig die zuvor zurückgegebenen Daten auf die lokale Festplatte (z.B. DAS). Wenn die Netzwerklatenz gering ist (wie es bei Rechenzentren der Fall ist) und das System konfliktfrei ist, kann es entweder den Speicher-Server (z.B. NAS) oder die lokale Festplatte (z.B. DAS) zu dessen vollständiger Nutzung ansteuern.

[0035] Eine Richtlinie kann festgelegt werden, um den Beginn des Vorablesezugriffs zu steuern. Beispielsweise kann der Vorablesezugriff beim Anwendungsfall einer schnellen VM-Erstellung so konfiguriert sein, dass er 12 Stunden nach der VM-Erstellung startet, so dass bei kurzlebigen VMs kein Vorablesezugriff durchgeführt wird. Bei einer VM-Migration kann der Vorablesezugriff so konfiguriert sein, dass er sofort nach der VM-Migration startet, so dass die Migration des Abbilds an der virtuellen Festplatte früher beendet werden kann.

[0036] Das ODS der vorliegenden Offenbarung kann gemäß einer weiteren Ausführungsform bei der Live--VM-Migration verwendet werden. In einer Cloud-Umgebung beispielsweise verbessert die Funktion der Live--VM-Migration die Cloud-Wartung und den Cloud-Betrieb erheblich. Mit der Zeit ist es beispielsweise unvermeidbar, dass ein Host einer Hardware-Wartung (z.B. Ersetzen eines unzuverlässigen CPU-Lüfters) oder einer Software-Wartung (z.B. Ausführen eines Sicherheits-Patches für den Hypervisor; bei KVM ist der Hypervisor Linux) unterzogen wird. Einige Wartungsvorgänge machen einen Neustart des Host erforderlich und können zu Ausfallzeiten von im Host ausgeführten VMs führen. Dank der Funktion der Live-VM-Migration können die betroffenen VMs auf andere Hosts migriert werden, bevor der Wartungsvorgang startet, so dass es zur keiner für den Benutzer wahrnehmbaren Ausfallzeit kommt.

[0037] Um eine VM von einem Quell-Host auf einen Ziel-Host zu migrieren, erfordern alle bestehenden Hypervisoren (darunter KVM, Xen und VMware), dass die Abbild-Datei der VM in einem gemeinsam genutzten Speicher gespeichert wird, auf den sowohl der Quell-Host als auch der Ziel-Host zugreifen kann. Sogar bei DAS ist es immer noch möglich, die Abbild-Datei der VM im Quell-Host für den Ziel-Host zugänglich zu machen (z.B. durch NFS), so dass die VM-Migration erfolgreich abgeschlossen werden kann. In diesem Fall werden jedoch alle von der VM im Ziel-Host erzeugten Festplatten-E/A-Anfragen an den Quell-Host geleitet und von diesem verarbeitet. Wird der Quell-Host neu gebootet, steht die Abbild-Datei nicht mehr zur Verfügung und die im Ziel-Host ausgeführte VM versagt.

[0038] Das ODS der vorliegenden Offenbarung unterstützt bei einer Ausführungsform die Live-Migration einer VM, die auf einer lokalen Festplatte (z.B. DAS) eines Host ausgeführt wird. **Fig. 5** zeigt Systemkomponenten und einen Betriebsablauf für eine Live-Migration einer virtuellen Maschine mit einem lokalen Speicher durch das ODS der vorliegenden Offenbarung. Ein Quell-Host ist ein physischer Computer oder eine physische Maschine und führt eine VM 508 unter Verwendung des ODS-Abbilds 504 der VM auf deren lokaler Festplatte 510 aus. Ein Ziel-Host 506 ist ein weiterer physischer Computer oder eine weitere physische Maschine, auf den bzw. die die VM 508 migriert wird. Der besseren Erläuterung wegen wird die VM 508 im Quell-Host 502 als Quell-VM bezeichnet; die VM im Ziel-Host 506 wird als Ziel-VM 512 bezeichnet. Vor Beginn der Migration exportiert der Quell-Host 502 die Abbild-Datei 504 der VM über NFS auf den Ziel-Host, so dass der Ziel-Host über das Netzwerk auf die Abbild-Datei zugreifen kann. Der Ziel-Host 506 erstellt auf seiner lokalen Festplatte 514 (z.B. DAS) ein ODS-Abbild 516 mit einem Kopfzeilenfeld der Metadaten des ODS-Abbilds, das auf die vom Quell-Host exportierte Abbild-Datei als Sicherungsabbild verweist oder zeigt. Die VM 508 kann dann sofort migriert werden, ohne dass die Abbild-Datei 504 der VM tatsächlich vom Quell--Host 502 auf den Ziel-Host 506 kopiert werden muss. Das heißt, dass die Ziel-VM 512 im Ziel-Host 506 unter ausschließlicher Verwendung des ODS-Abbilds 516 gebootet werden kann, das zunächst nur einen Verweis auf ein Sicherungsabbild beinhaltet. Während die Ziel-VM 512 ununterbrochen im Ziel-Host 506 ausgeführt wird, ermöglicht das ODS der vorliegenden Offenbarung gemäß einer Ausführungsform, dass die Ziel-VM 512 Daten durch Erstellen einer Kopie beim Lesen einbringt. Darüber hinaus führt das ODS der vorliegenden Offenbarung gemäß einer Ausführungsform einen Vorablesezugriff durch, um den Rest des Abbilds vom Quell-Host 502 auf den Ziel-Host 506 zu kopieren. Nach abgeschlossenem Vorablesezugriff wird die Abbild--Datei 504 der VM vollständig auf den Ziel-Host 506 kopiert und die Ziel-VM 512 ist in keiner Weise mehr vom Quell-Host 502 abhängig. Der Quell-Host 502 kann dann zu Wartungszwecken heruntergefahren werden.

[0039] Bei den oben beschriebenen ODS-Vorgängen für die Live-Migration von virtuellen Maschinen mit lokalem Speicher kann eine VM sofort migriert werden, ohne dass Daten im lokalen Speicher bewegt werden. Eine migrierte VM (Ziel-VM) kann weiterhin am neuen Speicherort ausgeführt werden, und Daten können nach Bedarf der Ziel-VM kopiert werden. Die von der Ziel-VM erzeugten Schreibvorgänge werden lokal gespeichert. Die Leerlaufzeit für Ressourcen wird genutzt, um im Ausgangs-Host (Quell-Host) zurückgelassene Daten vorab zu lesen. Nachdem alle Daten vom Quell-Host auf den Ziel-Host vorab gelesen wurden, wird die Migration abgeschlossen.

[0040] **Fig. 6** ist ein Ablaufplan, der ein Verfahren für das bedarfsgesteuerte Streaming von Abbildern virtueller Maschinen gemäß einer Ausführungsform zum Erstellen einer neuen VM zeigt. Bei 602 kann das Verfahren das Kopieren von mit einer ausgewählten virtuellen Maschine verknüpften Abbild-Metadaten von einem Speicher-Server in den lokalen Speicher eines Host-Computers beinhalten, auf dem eine oder mehrere Abbildvorlagen gespeichert sind, die jeweils einer oder mehreren virtuellen Maschinen entsprechen. Der lokale Speicher des Host-Computers beinhaltet zunächst keine Abbildvorlage der ausgewählten virtuellen Maschine. Bei 604 kann das Verfahren das Booten der ausgewählten virtuellen Maschine im Host-Computer unter Verwendung der kopierten Abbild-Metadaten beinhalten. Bei 606 kann das Verfahren das Ermöglichen beinhalten, dass die ausgewählte virtuelle Maschine Daten aus der Abbildvorlage auf dem Speicher-Server liest, die erforderlich sind, um das Ausführen der ausgewählten virtuellen Maschine im Host-Computer fortzu-

setzen, wenn die benötigten Daten nicht im lokalen Speicher des Host-Computers gespeichert sind. Bei 608 kann das Verfahren das Kopieren der gelesenen Daten der Abbildvorlage vom Speicher-Server in den lokalen Speicher des Host-Computers beinhalten, wenn die gelesenen Daten der Abbildvorlage nicht im lokalen Speicher des Host-Computers gespeichert sind. Nachfolgende Lesevorgänge der gleichen Daten werden aus dem lokalen Speicher des Host-Computers durchgeführt. Bei 610 kann das Verfahren auch das Setzen eines Bit in eine Bitmap beinhalten, um anzuzeigen, dass die gelesenen Daten im lokalen Speicher des Host-Computers gespeichert sind. Bei 612 werden Datenschreibvorgänge in die Abbildvorlage durch die ausgewählte virtuelle Maschine in den lokalen Speicher des Host-Computers geschrieben. Bei 614 kann die Ressourcen-Leerlaufzeit verwendet werden, um mit der ausgewählten virtuellen Maschine verknüpfte Daten der Abbildvorlage aus dem Speicher-Server in den lokalen Speicher des Host-Computers vorab zu lesen. In einem weiteren Aspekt können die Reaktionszeiten während des Abbild-Vorablesezugriffs überwacht werden, und der Vorablesezugriff kann vorübergehend angehalten werden, wenn eine lange Reaktionszeit ermittelt wird, beispielsweise wenn eine Reaktionszeit einen Grenzwert überschreitet.

[0041] In einem Aspekt beinhalten die Abbild-Metadaten, die vom Speicher-Server kopiert werden, zunächst einen Verweis auf die Abbildvorlage. Die Abbild-Metadaten können im lokalen Speicher des Host-Computers um eine Bitmap erweitert werden, die ein Bit einem entsprechenden Sektor der Abbildvorlage zuordnet. In einem weiteren Aspekt können die vom Speicher-Server kopierten Abbild-Metadaten neben dem Verweis auf die Abbildvorlage auch diese Bitmap beinhalten. Das entsprechende Bit in der Bitmap für die zur Ausführung der virtuellen Maschine erforderlichen Daten wird überprüft, um zu ermitteln, ob die zur Ausführung der ausgewählten virtuellen Maschine benötigten Daten der Abbildvorlage im lokalen Speicher des Host-Computers gespeichert sind. Je nach Bit in der Bitmap liest die ausgewählte virtuelle Maschine die Abbildvorlage im Speicher-Server oder die kopierte Abbildvorlage im lokalen Speicher des Host-Computers. Der Speicher-Server und der Host-Computer können Computer in einer Cloud-Umgebung sein, wobei beispielsweise eine Vielzahl von virtuellen Maschinen für Clients in der Cloud installiert ist.

[0042] Fig. 7 ist ein Ablaufplan, der ein Verfahren für ein bedarfsgesteuertes Streaming von virtuellen Maschinen zeigt, beispielsweise für eine VM-Live-Migration. Bei 702 kann das Verfahren das Kopieren von mit einer virtuellen Maschine verknüpften Abbild-Metadaten von einem Quell-Computer auf einen Ziel-Computer beinhalten, auf dem eine Abbildvorlage gespeichert ist, die der virtuellen Maschine entspricht. Der Ziel--Computer enthält die Abbildvorlage der virtuellen Maschine zunächst nicht. Bei 704 kann das Verfahren das Booten der virtuellen Maschine im Ziel-Computer unter dem Verwenden der kopierten Abbild-Metadaten beinhalten. Bei 706 kann das Verfahren das Ermöglichen beinhalten, dass die virtuelle Maschine im Ziel--Computer Daten der Abbildvorlage im Quell-Computer liest, die erforderlich sind, um das Ausführen der virtuellen Maschine im Ziel-Computer fortzusetzen, wenn die erforderlichen Daten der Abbildvorlage nicht im Ziel-Computer gespeichert sind. Bei 708 kann das Verfahren das Kopieren der gelesenen Daten der Abbildvorlage vom Quell-Computer auf den Ziel-Computer beinhalten, wenn die gelesenen Daten der Abbildvorlage nicht im Ziel-Computer gespeichert sind. Die nachfolgenden Lesevorgänge der gleichen Daten werden aus den kopierten Daten im Ziel-Computer durchgeführt. Bei 710 kann das Verfahren das Setzen eines Bits in eine Bitmap beinhalten, um anzuzeigen, dass die gelesenen Daten im Ziel-Computer gespeichert sind. Bei 712 werden Datenschreibvorgänge der virtuellen Maschine in den Ziel-Computer geschrieben. Bei 714 kann die Ressourcen-Leerlaufzeit genutzt werden, um mit der virtuellen Maschine verknüpfte Daten der Abbildvorlage aus dem Quell-Computer in den Ziel-Computer vorab zu lesen.

[0043] In einem Aspekt beinhalten die Abbild-Metadaten, die vom Quell-Computer kopiert werden, zunächst einen Verweis auf die Abbildvorlage. Die Abbild-Metadaten können im Ziel-Computer um eine Bitmap erweitert werden, die ein Bit einem entsprechenden Sektor der Abbildvorlage zuordnet. In einem weiteren Aspekt können die vom Quell-Computer kopierten Abbild-Metadaten neben dem Verweis auf die Abbildvorlage auch diese Bitmap beinhalten. Das entsprechende Bit in der Bitmap für die zur Ausführung der virtuellen Maschine erforderlichen Daten wird überprüft, um zu ermitteln, ob die zur Ausführung der virtuellen Maschine benötigten Daten im Ziel-Computer gespeichert sind. Je nach Bit in der Bitmap liest die virtuelle Maschine die Abbildvorlage im Quell-Computer oder die kopierte Abbildvorlage im Ziel-Computer. Das in **Fig. 7** gezeigte Verfahren kann für eine Live-Migration der virtuellen Maschine direkt vom Quell-Computer auf den Ziel-Computer durchgeführt werden, ohne dass ein separater Speicher-Server verwendet werden muss. Die VM kann sofort migriert werden, ohne dass ihre Abbild-Datei migriert wird. Während die VM ununterbrochen im Ziel-Host ausgeführt wird, kann das ODS der vorliegenden Offenbarung gemäß einer Ausführungsform einen Vorgang des Erstellens einer Kopie beim Lesen und einen Vorablesezugriff durchführen, um die Abbild-Datei schrittweise vom Quell-Host auf den Ziel-Host zu migrieren. Nach abgeschlossenem Vorablesezugriff ist die VM nicht länger vom Quell-Host abhängig, der für Wartungszwecke heruntergefahren werden kann.

[0044] In einem Aspekt können die Merkmale des ODS (Erstellen einer Kopie beim Schreiben, Erstellen einer Kopie beim Lesen und Vorablesezugriff) einzeln aktiviert werden, um unterschiedliche Anwendungsfälle zu unterstützen. Die **Fig. 8A** bis **Fig. 8D** zeigen mögliche unterschiedliche Anwendungsfälle unterschiedlicher Merkmale des ODS der vorliegenden Offenbarung gemäß einer Ausführungsform. In **Fig. 8A** ist im ODS nur die Funktion des Erstellens einer Kopie beim Schreiben aktiviert. Diese einfachste Grundkonfiguration kann mit den folgenden Vorteilen als Ersatz für das bestehende COW-Abbildformat des QEMU verwendet werden: 1) ODS berücksichtigt die cache=none-Option und cache=writethrough-Option des QEMU und garantiert die Datenintegrität bei einem Stromausfall. 2) Der Bitmap-Abschnitt und der Festplattendaten-Abschnitt des ODS werden am Seitenrand korrekt ausgerichtet und bieten ausgezeichnete Leistung. Beim Setup von **Fig. 8**, das sowohl eine Cloud- als auch Nicht-Cloud-Umgebung sein kann, sind die Speicheroptionen für das ODS-Abbild und die Abbildvorlage flexibel. Beide können auf einer oder mehreren lokalen Festplatten (z.B. DAS) des Host gespeichert werden, beide können auf einem Speicher-Server (z.B. NAS) gespeichert werden oder eine wird auf einer oder mehreren lokalen Festplatten (z.B. DAS) des Host und die andere auf einem Speicher-Server (z.B. NAS) gespeichert.

[0045] Die Setups der **Fig. 8B** und **Fig. 8C** können sich für eine Cloud-Umgebung eignen, wobei eine Abbildvorlage auf einem Speicher-Server (z.B. NAS) und das ODS-Abbild auf einer lokalen Festplatte (z.B. DAS) des Host-Computers gespeichert wird. Sie unterstützen eine schnelle VM-Erstellung und können den Inhalt der Abbildvorlage schrittweise in das ODS-Abbild migrieren, um eine übermäßige Last im Netzwerk und im Speicher-Server (z.B. DAS) zu vermeiden.

[0046] **Fig. 8D** zeigt ein Setup, wobei die Abbildvorlage auf einem Speicher-Server (z.B. NAS) gespeichert wird, und die drei ODS-Abbilder auf einer lokalen Festplatte (z.B. DAS) eines Host-Computers gespeichert werden. Dieses Setup ermöglicht mehreren VMs, unterschiedliche ODS-Abbilder des Erstellens einer Kopie beim Schreiben zu verwenden und gleichzeitig eine einzelne Nur-Lese-Kopie des „ODS (Erstellen einer Kopie beim Lesen + Vorablesen)“-Abbilds gemeinsam zu nutzen, die im lokalen Speicher (z.B. DAS) einen Spiegel des Inhalts der Abbildvorlage bietet. Bei diesem Setup wird das mehrfache Kopieren des Inhalts der Abbildvorlage jeweils für eine andere VM von einem Remote-Netzwerkspeicher-Server (z.B. NAS) in einen lokalen Speicher (z.B. DAS) eines Host vermieden.

[0047] Die vorliegende Offenbarung stellt auch die Datenintegrität bereit. Man nehme an, dass die folgenden Ereignisse hintereinander auftreten: (1) die VM sendet eine Festplatten-Schreibanfrage; (2) der ODS-Treiber (nach gewisser Verarbeitung) bestätigt den erfolgreichen Abschluss des Schreibvorgangs; und (3) die Stromzufuhr des Host wird sofort unterbrochen. Nachdem die Stromzufuhr wiederhergestellt ist, sollte bei dem nächsten Lesevorgang der VM des gleichen Sektors der vor dem Ausfall geschriebene Inhalt bezogen werden.

[0048] Tabelle 1 (oben) zeigt ein beispielhaftes Layout für ein ODS-Abbild auf einer Festplatte gemäß einer Ausführungsform der vorliegenden Offenbarung. Das ODS der vorliegenden Offenbarung bewahrt gemäß einer Ausführungsform die Datenintegrität unabhängig vom Zeitpunkt des Stromausfalls. Beispielsweise wenn ein Vorgang des Erstellens einer Kopie beim Schreiben und ein Vorgang des Erstellens einer Kopie beim Lesen ausgeführt wird, kann das ODS der vorliegenden Offenbarung bei einer Ausführungsform die Festplattendaten und die Bitmap getrennt aktualisieren. Bei einem Stromausfall zwischen den beiden Aktualisierungen wird die Datenintegrität bei der vorliegenden Offenbarung gemäß einer Ausführungsform weiterhin nicht beeinträchtigt, wie im Folgenden erläutert.

[0049] Ein Bit in der Bitmap kann in einem von zwei Status vorliegen, $S_{in\_backing=0}$ oder $S_{in\_ods=1}$, was bedeutet, dass sich der Inhalt des entsprechenden Sektors im Sicherungsabbild bzw. im ODS-Abbild befindet. Ein Status eines Sektors kann sich nur von $S_{in\_backing}$ zu S_{in_ods} und niemals von S_{in_ods} zu $S_{in\_backing}$ ändern. Es gibt zwei Szenarien, die den Status eines Sektors von $S_{in\_backing}$ zu S_{in_ods} ändern können: Erstellen einer Kopie beim Schreiben und Erstellen einer Kopie beim Lesen.

[0050] Ein Vorgang des Erstellens einer Kopie beim Schreiben wird durchgeführt, wenn das ODS eine Festplatten-Schreibanfrage von der VM bearbeitet. Der Kürze wegen wird bei der folgenden Erörterung davon ausgegangen, dass die Schreibanfrage sich über zwei Festplattensektoren (d1, d2) erstreckt. Man nehme an, bit(d1) und bit(d2) bezeichnen die Status von d1 bzw. d2 in der Bitmap. Es sei ferner angenommen, dass vor dem Schreibvorgang bit(d1) = $S_{in\_backing}$ und bit(d2) = $S_{in\_backing}$ gilt. Andere Fälle, bei denen mehr Datensektoren und unterschiedliche Ausgangstatus berücksichtigt werden, können ähnlich wie im folgenden Beispiel analysiert werden.

[0051] Das Bearbeiten der Schreibanfrage beim ODS kann den folgenden Operationsablauf beinhalten:

- ODS-W1: Die VM gibt eine Schreibanfrage für zwei Sektoren (d1, d2) aus.
- ODS-W2: Der ODS-Treiber speichert d1 im disk_data-Abschnitt des ODS auf der Festplatte.
- ODS-W3: Der ODS-Treiber speichert d2 im disk_data-Abschnitt des ODS auf der Festplatte.
- ODS-W4: Der ODS-Treiber aktualisiert bit(d1) von $S_{in\_backing}$ auf S_{in_ods}.
- ODS-W5: Der ODS-Treiber aktualisiert bit(d2) von $S_{in\_backing}$ auf S_{in_ods}.
- ODS-W6: Der ODS-Treiber bestätigt den Abschluss des Schreibvorgangs gegenüber der VM.

[0052] Es sei angemerkt, dass bit(d1) und bit(d2) zum gleichen Sektor gehören können und ODS-W4 und ODS-W5 somit im Rahmen einer einzelnen Aktualisierung durchgeführt werden können. Für eine Worst-Case-Analyse sind ODS-W4 und ODS-W5 zu trennen.

[0053] Der Host kann nach jedem der obigen Schritte ausfallen. Die vorliegende Offenbarung zeigt, dass das ODS die Datenintegrität ausfallunabhängig bewahrt. Insbesondere führt das ODS im Vergleich zu möglichen Szenarien beim Ursprungsabbildformat zu keiner weiteren Erschwernis. Das heißt, dass die Datenintegrität beim ODS zumindest genauso gut wie die Datenintegrität im Ursprungsabbildformat ist.

[0054] Wenn die VM das Ursprungsabbildformat verwendet, beinhaltet das Bearbeiten dieser Festplattenschreibung den folgenden Operationsablauf:

- RAW-W1: Die VM gibt eine Schreibanfrage für zwei Sektoren (d1, d2) aus.
- RAW-W2: Der Ursprungs-Treiber (RAW driver) speichert d1 auf der Festplatte.
- RAW-W3: Der Ursprungs-Treiber speichert d2 auf der Festplatte.
- RAW-W4: Der ODS-Treiber bestätigt den Abschluss des Schreibvorgangs gegenüber der VM.

[0055] Vor dem Schreibvorgang der VM werden die „alten“ Inhalte von d1 und d2 im Sicherungsabbild gespeichert. Nach dem Schreibvorgang der VM werden deren „neue“ Inhalte im ODS-Abbild gespeichert. Die Ausfallsszenarien bei ODS im Einzelnen:

- Ausfall nach ODS-W1. In diesem Fall entspricht das Verhalten des ODS einem Stromausfall beim Ursprungsabbildformat nach RAW-W1. Dadurch wird erzielt, dass der Schreibvorgang einfach verloren geht, was ein zulässiges, korrektes Verhalten ist, da der Treiber den Abschluss des Schreibvorgangs gegenüber der VM noch nicht bestätigt hat.

- Ausfall nach ODS-W2. In diesem Fall wird d1 in das ODS-Abbild geschrieben, aber bit(d1) wird nicht aktualisiert und bleibt $S_{in\_backing}$. Nachdem die Stromzufuhr wiederhergestellt ist, wird der Inhalt des nächsten Lesevorgangs der VM von d1 aus dem Sicherungsabbild bezogen, als würde der neue d1-Inhalt im ODS-Abbild nicht bestehen. Dieses Verhalten ist korrekt und entspricht einem Stromausfall beim Ursprungsabbildformat nach RAW-W1. Dadurch wird erzielt, dass der Schreibvorgang einfach verloren geht, was ein zulässiges, korrektes Verhalten ist, da der Treiber den Abschluss des Schreibvorgangs gegenüber der VM noch nicht bestätigt hat.

- Ausfall nach ODS-W3. Ähnlich wie oben bezieht der nächste Lesevorgang der VM von d1 oder d2 nach Wiederherstellung der Stromzufuhr den alten Inhalt aus dem Sicherungsabbild, als würde der neue Inhalt im ODS-Abbild nicht bestehen. Dieses Verhalten ist korrekt und entspricht einem Stromausfall beim Ursprungsabbildformat nach RAW-W1.

[0056] Ausfall nach ODS-W4. Nachdem die Stromzufuhr wiederhergestellt ist, bezieht der nächste Lesevorgang der VM von d1 dessen neuen Inhalt aus dem ODS-Abbild, während der nächste Lesevorgang der VM von d2 dessen alten Inhalt aus dem Sicherungsabbild bezieht (weil bit(d1) = S_{in_ods} und bit(d2) = $S_{in\_backing}$). Dieses Verhalten ist korrekt und entspricht einem Stromausfall beim Ursprungsabbildformat nach RAW-W2.

- Ausfall nach ODS-W5. Nachdem die Stromzufuhr wiederhergestellt ist, bezieht der nächste Lesevorgang der VM von d1 oder d2 den neuen Inhalt aus dem ODS-Abbild (weil bit(d1) = S_{in_ods} und bit(d2) = S_{in_ods}). Dieses Verhalten ist korrekt und entspricht einem Stromausfall beim Ursprungsabbildformat nach RAW-W3, d.h., der Schreibvorgang wird abgeschlossen, aber noch nicht bestätigt.

• Ausfall nach ODS-W6. Nachdem die Stromzufuhr wiederhergestellt ist, bezieht der nächste Lesevorgang der VM von d1 oder d2 den neuen Inhalt aus dem ODS-Abbild (weil bit(d1) = S_{in_ods} und bit(d2) = S_{in_ods}). Dieses Verhalten ist korrekt und entspricht einem Stromausfall beim Ursprungsabbildformat nach RAW-W4.

[0057] Die obige Analyse belegt, dass das ODS der vorliegenden Offenbarung gemäß einer Ausführungsform die Datenintegrität während eines Vorganges des Erstellens einer Kopie beim Schreiben bewahren kann. Mit einem ähnlichen Verfahren kann belegt werden, dass das ODS der vorliegenden Offenbarung bei einer Ausführungsform auch die Datenintegrität während eines Vorganges des Erstellens einer Kopie beim Lesen bewahren kann, indem die korrekte Aktualisierungsabfolge eingehalten wird - zuerst wird der Festplattendaten-Abschnitt und danach der Bitmap-Abschnitt des ODS-Abbilds aktualisiert.

[0058] Eine Umsetzung des ODS der vorliegenden Offenbarung kann gemäß einer Ausführungsform den Festplatten-Eingabe-/Ausgabe-(E/A-)Verarbeitungsaufwand verringern. Beispielsweise kann eine unbefangene Umsetzung des ODS im Vergleich zum Ursprungsabbildformat Verarbeitungsaufwand im Zusammenhang mit dem Lesen und Schreiben der Bitmap erzeugen. Im Worst-Case-Szenario kann ein Satz sequentieller Schreibanfragen der VM die nachstehende Schreibabfolge im Dateisystem des Host bewirken: Schreiben von s1, Schreiben von s2, Schreiben von bit(s2), Schreiben von s3, Schreiben von bit(s3) ... usw. Hier sind s1, s2 und s3 Sektoren mit einer aufeinanderfolgenden logischen Blockadresse, und bit(si) ist das entsprechende Bit von si in der Bitmap. Bei diesem Beispiel kann sich der Festplattenkopf zwischen dem Festplattendaten-Abschnitt und dem Bitmap-Abschnitt des ODS-Abbilds hin- und her bewegen. Gemäß einer Ausführungsform der vorliegenden Offenbarung wird eine Technik dargeboten, mit der der Verarbeitungsaufwand im Zusammenhang mit dem Aktualisieren der Bitmap in den meisten Fällen umgangen wird, ohne die Datenintegrität zu beeinträchtigen.

[0059] In einer Cloud-Umgebung ist die Größe der Abbildvorlage einer VM für gewöhnlich um einiges kleiner als der flüchtige Speicherplatz, der der VM zugewiesen ist. Beispielsweise sind 10 GB die maximal zulässige Abbildvorlagengröße für eine bekannte, im DAS ausgeführte VM, während sich der für diese VM bereitgestellte flüchtige Speicherplatz auf 170 GB oder mehr beläuft. Der zusätzliche flüchtige Speicherplatz kann für die VM entweder durch Erweitern der Stammfestplatte der VM auf Grundlage der Abbildvorlage oder durch Anschließen von zusätzlichen virtuellen Festplatten an die VM bereitgestellt werden. Eine weitere bekannte Cloud zielt auf Unternehmenskunden ab und bietet flexiblere Konfigurationen. Sie ermöglicht einer VM, eine Stammfestplatte mit einer Größer von viel mehr als 10 GB zu verwenden.

[0060] Im Folgenden wird das Verfahren zum Erstellen einer Linux-Abbildvorlage in einer Cloud beschrieben. Die Abbildvorlage verwendet das Ursprungsabbildformat. Es wird von einer anfänglichen Abbildvorlagengröße von 50 GB ausgegangen. Zunächst wird sie mit der erforderlichen Software installiert und vollständig getestet. Danach wird die Größe des ext3-Dateisystems in der Abbildvorlage mithilfe des resize2fs-Tools auf dessen Mindestgröße geändert (z.B. von 50 GB auf 12 GB). Die Abbildvorlage wird schließlich gekürzt, um der Mindestgröße des Dateisystems zu entsprechen (z.B. von 50 GB auf 12 GB). Beim Größenänderungs- und Kürzungsschritt werden überflüssige Daten entfernt, die während der Installation und des Testvorgangs erzeugt wurden, und es wird eine Abbildvorlage in Mindestgröße produziert. Eine kleine Abbildvorlage hilft dabei, die Menge an vom NAS in den DAS übertragenen Daten beim Erstellen von neuen VMs auf Grundlage der Abbildvorlage zu verringern.

[0061] Gemäß dem obigen Beispiel kann die 12-GB-Abbildvorlage verwendet werden, um VMs zu erstellen, deren Stammfestplattengröße variieren kann, abhängig davon, wie viel der Kunde bezahlt. Beispielsweise erstellt der folgende QEMU-Befehl ein 100-GB-ODS-Abbild im DAS auf Grundlage der im NFS gespeicherten 12-GB-Abbildvorlage.

qemu – img create – f ods – b /nfs/template.raw vm.ods 100*G*

[0062] Nachdem die Partitionsgröße der virtuellen Festplatte mit fdisk von 12 GB auf 100 GB erweitert wurde, kann resize2fs verwendet werden, um das ext3-Dateisystem im ODS-Abbild von 12 GB auf 100 GB zu erweitern, das zum großen Stammdateisystem der VM wird. Es sei angemerkt, dass das Verwenden von resize2fs zur Erweiterung eines Dateisystems ein schneller Vorgang ist, da dabei keine Neuanordnung von Blöcken erforderlich ist.

[0063] Beim in Tabelle 1 gezeigten ODS-Abbildformat ist die Festplattendatengröße die Größe des ODS-Abbilds, die von der VM wahrgenommen wird, und die effektive Sicherungsabbildgröße ist die Größe des Siche-

rungsabbilds. Beim obigen Beispiel ist die Festplattendatengröße = 100 GB und die effektive Sicherungsabbildgröße = 12 GB.

[0064] **Fig. 2** zeigt das Konzept, bei dem das ODS-Abbild größer als das Sicherungsabbild sein kann. Im Fall eines Datensektors, dessen logische Blockadresse (LBA) über die Größe des Sicherungsabbilds hinaus geht, kann dessen Inhalt nicht im Sicherungsabbild, sondern nur im ODS-Abbild vorhanden sein. Da der Status des Sektors von vornherein bekannt ist, muss der Status des Sektors nicht im Bitmap-Abschnitt des ODS-Abbilds erfasst werden. Folglich ist die Größe der Bitmap proportional zur Größe des Sicherungsabbilds und nicht zur Größe des ODS-Abbilds.

[0065] Bei einem 2-TB-ODS-Abbild, das auf ein 10-GB-Sicherungsabbild zeigt, beläuft sich die Größe der Bitmap lediglich auf 2,5 MB. Die gesamte Bitmap kann aufgrund ihrer geringen Größe leicht im Arbeitsspeicher zwischengespeichert werden, wodurch der Verarbeitungsaufwand im Zusammenhang mit dem wiederholten Lesen der Bitmap aus der Festplatte vermieden wird. Bei einer bekannten Cloud sind 10 GB die maximal zulässige Abbildvorlagengröße für eine VM, die in einem DAS ausgeführt wird. Bei Bearbeitung einer Leseanfrage von der VM für einen Sektor S, dessen logische Blockadresse (LBA) über die Größe des Sicherungsabbilds hinaus geht, weiß der ODS-Treiber lediglich auf der Grundlage der LBA, dass dieser Sektor nicht im Sicherungsabbild vorhanden sein kann und liest ihn folglich aus dem Festplattendaten-Abschnitt des ODS-Abbilds. Bei Bearbeitung einer Schreibanfrage von der VM für den Sektor S schreibt der ODS-Treiber die Daten direkt in den Festplattendaten-Abschnitt und die Bitmap muss nicht aktualisiert werden (tatsächlich gibt es nicht einmal die entsprechenden Bits für Sektor S in der Bitmap).

[0066] Da die Abbildvorlage mithilfe von resize2fs auf ihre Mindestgröße verringert wurde und die Daten in der Abbildvorlage, weil es sich um eine Vorlage handelt, vorwiegend Nur-Lese-Zugriff bieten (z.B. programmausführbar sind), zielen die meisten Festplatten-Schreibanfragen der VM auf jene Sektoren ab, deren Adressen über die Größe des Sicherungsabbilds hinaus gehen. Bei diesen Schreibanfragen schreibt der ODS-Treiber die Daten direkt auf eine lokale Festplatte (z.B. DAS) und es entsteht kein Verarbeitungsaufwand im Zusammenhang mit der Aktualisierung der Bitmap.

[0067] Im Folgenden wird eine Optimierung für Abbildvorlagen mit freien Bereichen, bei denen viele Datensektoren mit Nullen gefüllt sind, gemäß einer Ausführungsform der vorliegenden Offenbarung beschrieben. Für eine -Ursprungsabbildvorlage image.raw kann das qemu-img-Tool verwendet werden, um eine ODS-Abbildvorlage image.ods zu erstellen, deren Sicherungsabbild image.raw ist. Die Größe von image.ods entspricht der Größe von image.raw. Beim Erstellen von image.ods kann qemu-img nach mit Nullen gefüllten Sektoren S suchen und deren Status in der Bitmap auf S_{in_ods} setzen. Die Status für Sektoren ohne Nullen werden auf $S_{in\_backing}$ gesetzt. Zur Laufzeit, wenn die VM den Sektor S liest, dessen Status S_{in_ods} ist, liest der ODS-Treiber aus dem Festplattendaten-Abschnitt des ODS-Abbilds und erhält einen mit Nullen gefüllten Sektor zurück, wobei dies das gewünschte Verhalten ist. Dies geschieht, weil das ODS-Abbild im Dateisystem des Host als Datei mit freien Bereichen gespeichert wird und der Sektor S noch nie geschrieben wurde, und somit gibt das Betriebssystem (BS) des Host einen mit Nullen gefüllten Sektor zurück.

[0068] Die ODS-Abbildvorlage image.ods wird gemeinsam mit image.raw im NAS gespeichert. Beim Erstellen einer neuen VM auf einem Host kopiert es image.ods von einem Speicher-Server (z.B. NAS) auf eine lokale Festplatte (z.B. DAS) und ändert die Größe von image.ods auf die größere Zielgröße. Das Kopieren von image.ods geht schnell, da dessen Festplattendaten-Abschnitt leer und die Größe von image.ods somit klein ist. Insbesondere ist image.ods für ein 10-GB-image.raw nur ungefähr 2,5 MB groß. Das Ändern der Größe von image.ods in eine größere virtuelle Festplatte kann nur das Aktualisieren des disk_data_size-Felds in Tabelle 1 auf einen größeren Wert beinhalten. Wenn die VM bootet, kürzt der ODS--Treiber den Festplattendaten-Abschnitt automatisch auf die im disk_data_size-Feld angegebene Größe.

[0069] Wenn eine VM bootet, lädt der ODS-Treiber den Bitmap-Abschnitt der ODS-Abbilder von der Festplatte in den Arbeitsspeicher, der die Ausgangsstatus der Datensektoren enthält. Bei der vorliegenden Offenbarung werden diese beiden Bitmap-Kopien als On-Disk-Status bzw. In-Memory-Status bezeichnet. Zur Laufzeit hält der ODS-Treiber den In-Memory-Status stets auf dem neuesten Stand, kann den On-Disk-Status hingegen allmählich aktualisieren, um den Festplatten-E/A-Verarbeitungsaufwand zu verringern. Bei einem Stromausfall ist jedoch gewährleistet, dass veraltete Informationen im On-Disk-Status die Datenintegrität niemals beeinträchtigen.

[0070] Wenn die VM einen Sektor liest, dessen In-Memory-Status S_{in_ods} ist, liest der ODS-Treiber den Sektor aus dem ODS-Abbild und gibt ihn an die VM zurück. Es ist kein weiterer Verarbeitungsaufwand damit ver-

bunden. Wenn die VM einen Sektor liest, dessen In-Memory-Status $S_{in\_backing}$ ist, liest der ODS-Treiber den Sektor aus dem Sicherungsabbild und gibt ihn an die VM zurück. Nachdem die VM die Verarbeitung der zurückgegebenen Daten im Hintergrund fortgesetzt hat (d.h. asynchron), schreibt der ODS-Treiber den Inhalt des Sektors in den Festplattendaten-Abschnitt des ODS-Abbilds und aktualisiert den In-Memory-Status des Sektors von $S_{in\_backing}$ auf S_{in_ods}. Der On-Disk-Status des Sektors wird jedoch nicht aktualisiert und bleibt $S_{in\_backing}$, wodurch der Festplatten-E/A-Verarbeitungsaufwand verringert wird. Die „genutzten“ Bits des In-Memory-Status können gelöscht werden, um den On-Disk-Status allmählich zu aktualisieren, entweder periodisch (z.B. einmal stündlich) oder wenn die VM herunterfährt. Wenn die Stromzufuhr des Host unterbrochen wird, bevor der On-Disk-Status aktualisiert wurde, lädt der ODS-Treiber nach Wiederherstellung der Stromzufuhr den veralteten On-Disk-Status erneut in den Arbeitsspeicher und der Status des Sektors wird als $S_{in\_backing}$ beibehalten. Bei Bearbeitung des nächsten Lesevorgangs der VM dieses Sektors wiederholt der ODS-Treiber den Vorgang des Erstellens einer Kopie beim Lesen erneut: Lesen des Sektors aus dem Sicherungsabbild, Zurückgeben des Sektors an die VM, asynchrones Schreiben des Sektors in das ODS-Abbild und Aktualisieren des In-Memory-Status von $S_{in\_backing}$ auf S_{in_ods}. Ohne den On-Disk-Status während des Vorgangs des Erstellens einer Kopie beim Schreiben sofort zu aktualisieren, bezieht die VM nach Wiederherstellung dennoch den korrekten Sektorinhalt, auch wenn sie den bereits in das ODS-Abbild kopierten Inhalt des Sektors ignorieren kann.

[0071] Wenn die VM in einen Sektor schreibt, überprüft der ODS-Treiber den On-Disk-Status (im Gegensatz zum In-Memory-Status), um die adäquate Aktion zu ermitteln. Wenn der On-Disk-Status des Sektors S_{in_ods} ist (der In-Memory-Status des Sektors ist ebenfalls S_{in_ods}), schreibt der ODS-Treiber den Sektor direkt in das ODS-Abbild und bestätigt gegenüber der VM, dass der Schreibvorgang abgeschlossen ist. Es fällt kein Verarbeitungsaufwand im Zusammenhang mit der Bitmap-Aktualisierung an. Wenn der On-Disk-Status des Sektors $S_{in\_backing}$ ist (der In-Memory-Status des Sektors kann entweder $S_{in\_backing}$ oder S_{in_ods} sein), schreibt der ODS-Treiber den Sektor in das ODS-Abbild, aktualisiert den On-Disk-Status auf S_{in_ods} (und aktualisiert auch den In-Memory-Status auf S_{in_ods}, wenn er derzeit $S_{in\_backing}$ lautet) und bestätigt gegenüber der VM, dass der Schreibvorgang abgeschlossen ist. In diesem Fall kann es zu einem Verarbeitungsaufwand im Zusammenhang mit der Aktualisierung des On-Disk-Status kommen.

[0072] Bei einer weiteren Ausführungsform kann eine asynchrone Umsetzung bereitgestellt werden, die den Arbeitsspeicher-Verarbeitungsaufwand verringert. Alle Blockeinheitentreiber in QEMU setzen die BlockDriver-Schnittstelle um, die APIs für sowohl eine synchrone E/A als auch eine asynchrone E/A bereitstellt. Erstere ermöglicht der Blockeinheit lediglich, jeweils nur eine ausstehende E/A-Anfrage zu bearbeiten. Letzere ermöglicht der Blockeinheit, mehrere ausstehende E/A-Anfragen gleichzeitig zu bearbeiten, indem der Blockeinheitentreiber die VM mithilfe von Callback-Funktionen über den Abschluss der E/A-Vorgänge informiert.

[0073] Das ODS der vorliegenden Offenbarung setzt gemäß einer Ausführungsform die asynchrone Schnittstelle um. Bei wenigen Ausnahmefällen wird die Durchführung von Vorgängen des Erstellens einer Kopie beim Schreiben und des Erstellens einer Kopie beim Lesen im gleichen Datensektor mit Vorsicht gehandhabt. Man nehme an, dass die VM eine Leseanfrage R_d für einen Datensektor d sendet, dessen In-Memory--Status $S_{in\_backing}$ ist, und dann eine Schreibanfrage W_d für den gleichen Sektor d sendet, bevor der Lesevorgang abgeschlossen ist. Man nehme an, dass der ODS-Treiber bei Bearbeitung dieses Falls die Operationen in der nachstehenden Reihenfolge beendet:

1. Lesen des alten Inhalts des Sektors aus dem Sicherungsabbild im Rahmen des Vorganges des Erstellens einer Kopie beim Lesen für R_d.

2. Schreiben des neuen Inhalts des Sektors in das ODS-Abbild im Rahmen des Vorganges des Erstellens einer Kopie beim Schreiben für W_d.

3. Schreiben des alten Inhalts des Sektors in das ODS-Abbild im Rahmen des Vorganges des Erstellens einer Kopie beim Lesen für R_d.

[0074] Es kann zu einer Konkurrenzsituation kommen, wenn ein Vorgang des Erstellens einer Kopie beim Schreiben und ein Vorgang des Erstellens einer Kopie beim Lesen für den gleichen Datensektor ausgeführt werden. Im ODS-Abbild bleibt sodann der alte Inhalt des Sektors zurück, was ein falsches Ergebnis ist. Um diese und andere ähnliche Konkurrenzsituationen richtig handhaben zu können, überprüft der ODS-Treiber, bevor er einen Vorgangs des Erstellens einer Kopie beim Lesen für einen Datensektor d ausführt, ob für d ein ausstehender Vorgang des Erstellens einer Kopie beim Schreiben vorliegt. Ist dies der Fall, wird der Vorgang des Erstellens einer Kopie beim Lesen eingestellt. Gleichermaßen überprüft der ODS-Treiber, bevor er einen Vorgang des Erstellens einer Kopie beim Schreiben für einen Datensektor d ausführt, ob für d ein ausstehender Vorgang des Erstellens einer Kopie beim Lesen vorliegt. Ist dies der Fall, wird der Vorgang des

Erstellens einer Kopie beim Schreiben verzögert, bis der ausstehende Vorgang des Erstellens einer Kopie beim Lesen abgeschlossen wurde, wodurch gewährleistet wird, dass der neue Inhalt auf der Festplatte zurückbleibt.

[0075] Das ODS der vorliegenden Offenbarung kann gemäß einer Ausführungsform die folgenden Optimierungen bereitstellen:

- Die Größe des Bitmap-Abschnitts eines ODS-Abbilds ist proportional zur Größe des Sicherungsabbilds und nicht zur Größe des ODS-Abbilds. Bei einem 2-TB-ODS-Abbild, das auf ein 10-GB-Sicherungsabbild zeigt, beläuft sich die Größe der Bitmap lediglich auf 2,5 MB. Es sei angemerkt, dass 10 GB die maximal zulässige Abbildvorlagengröße für eine bekannte Cloud-VM ist, die in einem DAS ausgeführt wird.
- Aufgrund der kleinen Größe der Bitmap kann eine vollständige Kopie der Bitmap im Arbeitsspeicher behalten werden, um Verarbeitungsaufwand im Zusammenhang mit wiederholtem Lesen der On-Disk--Bitmap zu vermeiden.
- Nachdem der Vorablesezugriff abgeschlossen wurde, arbeitet das ODS-Abbild beinahe wie ein Ursprungsabbild. Festplatten-Lese- oder Festplatten-Schreibanfragen von der VM werden direkt im Vergleich zu dem Festplattendaten-Abschnitt des ODS-Abbilds ausgeführt, ohne dass Verarbeitungsaufwand im Zusammenhang mit dem Prüfen der In-Memory-Bitmap oder dem Aktualisieren der On-Disk--Bitmap entsteht.
- Bei Bearbeitung einer Lese- oder Schreibanfrage einer VM für einen Sektor, dessen logische Blockadresse über die Größe des Sicherungsabbilds hinaus geht, liest oder schreibt der ODS-Treiber den Festplattendaten-Abschnitt des ODS-Abbilds direkt, ohne dass Verarbeitungsaufwand im Zusammenhang mit dem Prüfen der In-Memory-Bitmap oder dem Aktualisieren der On-Disk-Bitmap entsteht.
- Bei Bearbeitung einer Leseanfrage einer VM für einen Sektor, dessen In-Memory-Status bereits S_{in_ods} lautet, wird der Sektor direkt aus dem ODS-Abbild gelesen, ohne dass Verarbeitungsaufwand im Zusammenhang mit dem Aktualisieren der On-Disk-Bitmap entsteht.
- Bei einem Vorgang des Erstellens einer Kopie beim Lesen wird nur die In-Memory-Bitmap aktualisiert und die On-Disk-Bitmap wird nicht sofort aktualisiert.
- Ein Vorgang des Erstellens einer Kopie beim Lesen ist nicht Teil des kritischen Pfads des Zurückgebens der Daten an die VM. Die Daten werden asynchron im Hintergrund im ODS-Abbild gespeichert, während die VM die Verarbeitung der aus dem Sicherungsabbild gelesenen Daten fortsetzt.
- Bei Bearbeitung einer Schreibanfrage einer VM für einen Sektor, dessen On-Disk-Status bereits S_{in_ods} lautet, entsteht kein Verarbeitungsaufwand im Zusammenhang mit der Aktualisierung der On-Disk-Bitmap.
- Wenn ein Sektor im Sicherungsabbild zur Gänze mit Nullen gefüllt ist, kann dessen Ausgangsstatus in der On-Disk-Bitmap des ODS-Abbilds auf S_{in_ods} gesetzt werden, so dass das Lesen oder Schreiben des Sektors so behandelt wird, als wäre der Sektor bereits im ODS-Abbild, und es entsteht kein Verarbeitungsaufwand im Zusammenhang mit dem Aktualisieren der On-Disk-Bitmap.
- Das Ändern der Größe eines ODS-Abbilds ist eine konstante Zeitoperation, bei der nur das Feld disk_data_size im in Tabelle 1 gezeigten Layout aktualisiert werden muss.

[0076] Die Konzepte des ODS der vorliegenden Offenbarung beinhalten gemäß einer Ausführungsform die Vorgänge des Erstellens einer Kopie beim Schreiben, des Erstellens einer Kopie beim Lesen sowie des Vorablesezugriffs. Theoretisch kann es möglich sein, die Vorgänge des Erstellens einer Kopie beim Lesen und des Vorablesezugriffs bei bestehenden Formaten des Erstellens einer Kopie beim Schreiben umzusetzen, die vom QEMU (z.B. CoW und QCOW2) bereits unterstützt werden, wodurch die Erschwernis des Einführens eines neuen Abbildformats vermieden wird. Das neue ODS-Abbildformat der vorliegenden Offenbarung erzielt gemäß einer Ausführungsform in den häufigsten Fällen eine hohe Leistung. Das Format des Erstellens einer Kopie beim Schreiben. Das COW-Format ist zum ODS-Format beinahe identisch. Es beinhaltet ebenfalls eine Kopfzeile, einen Bitmap-Abschnitt und einen Festplattendaten-Abschnitt. Bei der aktuellen Umsetzung des COW-Treibers werden die cache=none-Option und cache=writethrough-Option des QEMU ignoriert. Folglich können Festplattendaten bei einem Stromausfall korrumpiert werden. Diese Umsetzungsprobleme können potenziell behoben werden, aber das COW-Format selbst ist mit einer wesentlichen Einschränkung verbunden - sein Festplattendaten-Abschnitt ist nicht an der 4-KB-Seitenbegrenzung ausgerichtet. Sogar wenn die Lese- oder Schreibanfrage der VM an der 4-KB-Seitenbegrenzung der virtuellen Fest-

platte ausgerichtet ist, ist sie, nachdem die Anfrage übersetzt wurde, um im Festplattendaten-Abschnitt des COW-Abbilds zu arbeiten, womöglich nicht länger an der 4-KB-Seitenbegrenzung des Host-Dateisystems ausgerichtet. Da der Seiten-Cachespeicher des Host auf 4-KB-Seiten arbeitet, kann eine fehlausgerichtete Anfrage zu mehreren Festplatten-E/As im Host führen. Beispielsweise kann ein von der VM ausgegebener, gut ausgerichteter 4-KB-Schreibvorgang in einem fehlausgerichteten 4-KB-Schreibvorgang im Host übersetzt werden, wodurch ein ineffizientes Lesen-Ändern-Schreiben-Verhalten bewirkt wird, d.h., es werden 8 KB gelesen, 4 KB geändert und 8 KB zurückgeschrieben. Das ODS-Format der vorliegenden Offenbarung berücksichtigt dieses Problem bei einer Ausführungsform durch das Hinzufügen der in Tabelle 1 gezeigten Auffüllabschnitte, um zu gewährleisten, dass der Bitmap-Abschnitt und der disk- data-Abschnitt korrekt an der 4-KB-Seitenbegrenzung ausgerichtet sind.

[0077] QCOW2 ist das „native Format“ des QEMU. Es unterscheidet sich dahingehend wesentlich von COW und ODS, dass es, anstatt die Unterstützung der Host-Dateisysteme in Bezug auf Dateien mit freien Bereichen zu nutzen, seinen eigenen 2-Ebenen-Index umsetzt, um Abbild-Dateien mit freien Bereichen zu unterstützen. Der Index bildet eine logische Blockadresse in eine Position in der Abbild-Datei ab, in der der Inhalt des Blocks tatsächlich gespeichert ist. Dank dieser Flexibilität bei der Adressabbildung kann QCOW2 moderne Funktionen wie Schnappschuss und Komprimierung bereitstellen.

[0078] Andererseits führt der 2-Ebenen-Index von QCOW2 auch zu Verarbeitungsaufwand, insbesondere zu zusätzlichen Festplattenzugriffen im Zusammenhang mit dem Lesen oder Aktualisieren des Index. Im Vergleich zu den Optimierungen bei ODS weist eine potenzielle Umsetzung von QCOW2, die um einen Vorgang des Erstellens einer Kopie beim Lesen und einen Vorablesezugriff erweitert ist, die folgenden Einschränkungen auf:

- Die Größe des Index ist proportional zur Größe des (großen) QCOW2-Abbilds und nicht zur Größe des (kleinen) Sicherungsabbilds. Folglich kann der Index nicht zur Gänze im Arbeitsspeicher zwischengespeichert werden.
- Sogar nach abgeschlossenem Vorablesezugriff muss sie den On-Disk-Index immer noch lesen, um eine von der VM ausgegebene Leseanfrage zu bearbeiten, was zusätzliche Festplatten-E/A-Vorgänge bedeutet.
- Sogar nach abgeschlossenem Vorablesezugriff muss sie den On-Disk-Index immer noch lesen (und potenziell schreiben), um eine von der VM ausgegebene Schreibanfrage zu bearbeiten.
- Bei Bearbeitung einer Lese- oder Schreibanfrage einer VM für einen Sektor, dessen logische Blockadresse über die Größe des Sicherungsabbilds hinaus geht, muss sie den Index immer noch lesen oder schreiben.
- Bei Vorgängen des Erstellens einer Kopie beim Lesen und Vorablesezugriffen besteht die Wahrscheinlichkeit, dass der On-Disk-Index häufiger aktualisiert wird, da sie nicht den gesamten Index im Arbeitsspeicher halten kann.

[0079] Die Optimierungen beim ODS der vorliegenden Offenbarung umgehen den Verarbeitungsaufwand im Zusammenhang mit der Aktualisierung der On-Disk-Bitmap in den häufigsten Fällen. Im Gegensatz dazu kann mit dem 2-Ebenen-Index des QCOW2 nicht die gleiche Optimierungshöhe erzielt werden. Da die modernen Funktionen von QCOW2 (d.h. Schnappschuss, Komprimierung und Verschlüsselung) in einer Cloud nicht verwendet werden, kann die vorliegende Offenbarung bei einer Ausführungsform ein einfacheres Abbildformat vorsehen, das eine bessere Leistung bietet, d.h. das ODS-Format. Die Schnappschuss-Funktion von QCOW2 bietet die besten Voraussetzungen, um in einer Cloud nützlich zu sein. Eine Cloud stellt für gewöhnlich zwei schnappschussähnliche Funktionen bereit: 1) zuverlässiges Backup, und 2) abbildgetreues Zusammenpacken, d.h., Schießen eines Schnappschusses des Stammdateisystems, Umwandeln in eine Abbildvorlage und Verzeichnen der Abbildvorlage in der Cloud zur künftigen Wiederverwendung. Ein Schnappschuss von QCOW2 wird jedoch im QCOW2-Abbild im DAS gespeichert und ist somit kein zuverlässiger Backup-Mechanismus und kann nicht als Abbildvorlage zur Erstellung einer neuen VM auf einem anderen Host verwendet werden.

[0080] Die vorliegende Offenbarung kann bei einer Ausführungsform sowohl die schnelle Erstellung von virtuellen Maschinen (VM) als auch eine gute Laufzeitleistung in der Cloud unterstützen. Die bestehenden Lösungen weisen zumindest ein, zwei Beschränkungen auf: 1) langsame VM-Erstellung, da das gesamte Abbild vor Erstellung kopiert wird (z.B. Ursprungsabbildformat-Treiber bei KVM); 2) hoher Netzwerkdatenverkehr und schlechte Leistung zur Laufzeit, da Daten wiederholt aus dem Remote-Speicher-Server gelesen werden (z.B. qcow2-Abbildformat-Treiber bei KVM).

[0081] Bei einer Ausführungsform kann das bedarfsgesteuerte Streaming der vorliegenden Offenbarung sowohl eine schnelle VM-Bereitstellung als auch eine gute Laufzeitleistung vorsehen. Das ODS der vorliegenden Offenbarung kann einen Hypervisor um das neue „ods“-Abbildformat und den entsprechenden Treiber erweitern. Im Vergleich zum Ursprungsabbildformat, das bei einigen bestehenden Clouds verwendet wird, führt das ODS zu weniger Netzwerkdatenverkehr und einer geringeren Eingabe-/Ausgabe-(E/A-)Last im Speicher-Server, und das nicht nur während der Bereitstellung, sondern allgemein während der gesamten Lebensdauer der VM. Im Gegensatz zum Ursprungsabbildformat, das bei einigen bestehenden Clouds verwendet wird, kann das ODS eine VM über das Netzwerk booten, ohne dass eine vollständige Kopie der Abbildvorlage erstellt wird. Es kann eine VM sofort booten und danach Datenblöcke nach Bedarf aus dem Speicher-Server vorab lesen, wenn die VM auf die Datenblöcke zugreift.

[0082] Im Gegensatz zum QCOW2-Abbildformat, bei dem nur ein Vorgang des Erstellens einer Kopie beim Schreiben, aber kein Vorgang des Erstellens einer Kopie beim Lesen durchgeführt wird und somit der gleiche Datenblock wiederholt aus einem Speicher-Server gelesen werden kann, kann das ODS der vorliegenden Offenbarung gemäß einer Ausführungsform einen Datenblock höchstens einmal aus einem Speicher-Server lesen und diesen Block dann zur späteren Wiederverwendung auf einer lokalen Festplatte speichern. Ein weiterer Vorteil des ODS der vorliegenden Offenbarung gegenüber QCOW2 kann sein, dass sich das Daten--Layout von QCOW2 auf einer lokalen Festplatte von jenem des Ursprungsabbildformats unterscheidet; andererseits kann das Datenblock-Layout des ODS der vorliegenden Offenbarung in Bezug auf jenes des Ursprungsabbildformats identisch sein. Folglich kann die Laufzeitleistung des ODS besser als jene von QCOW2 sein.

[0083] Experimente zeigen, dass 1) das ODS der vorliegenden Offenbarung einen bekannten Betriebssystem-Server innerhalb von 14 Sekunden booten kann und dabei Daten in der Größe von weniger als 17 Megabyte (MB) über das Netzwerk überträgt; und 2) die Laufzeitleistung des ODS genauso gut wie das Ursprungsabbildformat ist.

[0084] Bei einer weiteren Ausführungsform kann das ODS der vorliegenden Offenbarung darüber hinaus eine moderne Funktion beinhalten, die das gesamte VM-Abbild im Hintergrund aus einem Speicher-Server vorab liest, wenn oder während sich die Ressourcen wie die Festplatte, das Netzwerk und die CPU ansonsten im Leerlauf befinden. Diese Funktion verdeckt Netzwerklatenz und verteilt die Ressourcennutzung gleichmäßig anstatt zu warten und das gesamte VM-Abbild während der VM-Erstellung zu kopieren.

[0085] Das ODS der vorliegenden Offenbarung kann in einer Cloud-Umgebung verwendet werden. Das ODS kann darüber hinaus in einer Nicht-Cloud-Umgebung verwendet werden. Ferner kann das ODS verwendet werden, wobei nur die Funktion des Erstellens einer Kopie beim Schreiben aktiviert ist, beispielsweise um als Hochleistungs-CoW-Format zu dienen. Außerdem kann die Umsetzung des ODS für die Gast-VM (ausgewählte VM, die auf einem Hypervisor oder dergleichen ausgeführt wird) transparent und somit umfassend anwendbar sein.

[0086] Das ODS der vorliegenden Offenbarung kann als Teil eines Hypervisors oder als eine Erweiterung einer Funktion eines Hypervisors oder dergleichen umgesetzt sein, der bzw. die beispielsweise alle Funktionen eines Hypervisors bereitstellt, ohne dass Änderungen an der Gast-VM vorgenommen werden. Die Fähigkeit des ODS, d.h. der Vorgang des Erstellens einer Kopie beim Schreiben, der Vorgang des Erstellens einer Kopie beim Lesen und Vorablesezugriff, wird von keinem bestehenden Server bereitgestellt.

[0087] Die Größe des Bitmap-Abschnitts eines ODS-Abbilds ist proportional zur Größe des Sicherungsabbilds und nicht zur Größe des ODS-Abbilds. Ferner kann aufgrund der geringen Größe der Bitmap eine vollständige Kopie der Bitmap im Arbeitsspeicher aufbewahrt werden, wodurch Verarbeitungsaufwand im Zusammenhang mit dem wiederholten Lesen der On-Disk-Bitmap vermieden wird. In einem Aspekt besteht die Möglichkeit, dass im Rahmen des Vorgangs des Erstellens einer Kopie beim Lesen nur die In-Memory--Bitmap aktualisiert wird und die On-Disk-Bitmap nicht sofort aktualisiert werden muss, wodurch der Festplatten-E/A-Verarbeitungsaufwand verringert werden kann.

[0088] Beim Bearbeiten einer Lese- und/oder Schreibanfrage einer VM für einen Sektor, dessen logische Blockadresse über die Größe des Sicherungsabbilds hinaus geht, liest und/oder schreibt der ODS-Treiber den Festplattendaten-Abschnitt des ODS-Abbilds direkt, ohne dass Verarbeitungsaufwand im Zusammenhang mit dem Prüfen der In-Memory-Bitmap und/oder dem Aktualisieren der On-Disk-Bitmap entsteht. Ferner werden Festplatten-Lese- und/oder Festplatten-Schreibanfragen von der VM nach Abschluss des Vorablesezugriffs direkt im Vergleich zu dem Festplattendaten-Abschnitt des ODS-Abbilds ausgeführt, ohne dass Ver-

arbeitungsaufwand im Zusammenhang mit dem Prüfen der In-Memory-Bitmap und/oder dem Aktualisieren der On-Disk-Bitmap entsteht.

[0089] In einem weiteren Aspekt besteht die Möglichkeit, dass der Vorgang des Erstellens einer Kopie beim Lesen nicht Teil des kritischen Pfads des Zurückgebens der Daten an die VM ist, und die Daten können asynchron im Hintergrund in das ODS-Abbild gespeichert werden, während die VM die Verarbeitung der aus dem Sicherungsabbild gelesenen Daten fortsetzt.

[0090] In einem noch weiteren Aspekt wird der Ausgangsstatus eines Datensektors, wenn der Datensektor im Sicherungsabbild zur Gänze mit Nullen gefüllt ist, in der On-Disk-Bitmap des ODS-Abbilds so eingestellt, als würde sich der Sektor bereits im ODS-Abbild befinden, wodurch der Verarbeitungsaufwand im Zusammenhang mit dem Aktualisieren der On-Disk-Bitmap und dem Lesen des Datensektors aus dem Speicher--Server vermieden wird.

[0091] Wie der Fachmann verstehen wird, können Aspekte der vorliegenden Erfindung in Form eines Systems, eines Verfahrens oder eines Computerprogrammprodukts umgesetzt sein. Demgemäß können Aspekte der vorliegenden Erfindung die Form einer ausschließlich aus Hardware bestehenden Ausführungsform, einer ausschließlich aus Software bestehenden Ausführungsform (Firmware, residente Software, Microcode usw. mit eingeschlossen) oder einer Ausführungsform annehmen, die Software- und Hardware-Aspekte kombiniert, die hier allesamt allgemein als „Schaltung“, „Modul“ oder „System“ bezeichnet werden können. Ferner können Aspekte der vorliegenden Erfindung die Form eines Computerprogrammprodukts annehmen, das als ein oder mehrere computerlesbare Medien umgesetzt ist, die einen computerlesbaren Programmcode aufweisen.

[0092] Es kann eine beliebige Kombination aus einem oder mehreren computerlesbaren Medien verwendet werden. Das computerlesbare Medium kann ein computerlesbares Signalmedium oder ein computerlesbares Speichermedium sein. Ein computerlesbares Speichermedium kann beispielsweise ein/e elektronische/s, magnetische/s, optische/s, elektromagnetische/s, Infrarot- oder Halbleitersystem, -vorrichtung oder -einheit oder eine geeignete Kombination des Vorstehenden sein, ohne jedoch darauf beschränkt zu sein. Spezifischere Beispiele (nichterschöpfende Liste) für das computerlesbare Speichermedium sind unter anderem: eine elektrische Verbindung mit einem oder mehreren Kabeln, eine tragbare Computerdiskette, eine Festplatte, ein Direktzugriffsspeicher (RAM), ein Nur-Lese-Speicher (ROM), ein elektronisch löschbarer programmierbarer Nur-Lese-Speicher (EPROM oder Flash-Speicher), ein Lichtwellenleiter, ein tragbarer Compact Disk-Nur-Lese-Speicher (CD-ROM), eine optische Speichereinheit, eine magnetische Speichereinheit oder eine geeignete Kombination des Vorstehenden. Im Kontext dieses Dokuments kann ein computerlesbares Speichermedium jedes konkrete Medium sein, das ein Programm zur Verwendung durch ein Anweisungsausführungssystem, eine -vorrichtung oder eine -einheit oder in Verbindung damit enthalten oder speichern kann.

[0093] Ein computerlesbares Signalmedium kann ein weitergeleitetes Datensignal beinhalten, das einen computerlesbaren Programmcode aufweist, beispielsweise im Basisband oder als Teil einer Trägerwelle. Ein solches weitergeleitetes Signal kann eine Vielzahl von Formen annehmen, beispielsweise elektromagnetisch, optisch oder eine geeignete Kombination davon, ohne jedoch darauf beschränkt zu sein. Ein computerlesbares Signalmedium kann ein beliebiges computerlesbares Medium sein, bei dem es sich nicht um ein computerlesbares Speichermedium handelt und das ein Programm zur Verwendung durch ein Anweisungsausführungssystem, eine - vorrichtung oder eine -einheit oder in Verbindung damit übertragen, weiterleiten oder transportieren kann.

[0094] Der in einem computerlesbaren Medium enthaltene Programmcode kann mithilfe eines geeigneten Mediums übertragen werden, beispielsweise drahtlos, kabelgebunden, Lichtwellenleiterkabel, HF usw. oder eine Kombination des Vorstehenden, ohne jedoch darauf beschränkt zu sein.

[0095] Ein Computerprogrammcode zum Ausführen von Operationen für Aspekte der vorliegenden Erfindung kann in irgendeiner Kombination aus einer oder mehreren Programmiersprachen geschrieben sein, beispielsweise objektorientierte Programmiersprache wie Java, Smalltalk, C++ oder dergleichen und herkömmliche prozedurale Programmiersprachen wie die „C“-Programmiersprache oder ähnliche Programmiersprachen, eine Skriptsprache wie Perl, VBS oder ähnliche Sprachen und/oder funktionelle Sprachen wie Lisp und ML und logikorientierte Sprachen wie Prolog. Der Programmcode kann zur Gänze 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 Remote-Computer oder zur Gänze auf dem Remote--

Computer oder -Server ausgeführt werden. Bei letzterem Szenario kann der Remote-Computer über einen beliebigen Netzwerktyp, beispielsweise Local Area Network (LAN) oder Wide Area Network (WAN), mit dem Computer des Benutzers verbunden sein oder die Verbindung zu einem externen Computer kann hergestellt werden (z.B. über einen Internet-Diensteanbieter über Internet).

[0096] Aspekte der vorliegenden Erfindung sind unter Bezugnahme auf die Ablaufplandarstellungen und/oder Blockschaubilder von Verfahren, Vorrichtungen (Systemen) und Computerprogrammprodukten gemäß Ausführungsformen der Erfindung beschrieben. Es versteht sich, dass jeder Block der Ablaufplandarstellungen und/oder Blockschaubilder und Kombinationen von Blöcken in den Ablaufplandarstellungen und/oder Blockschaubildern durch Computerprogrammanweisungen umgesetzt werden können. Diese Computerprogrammanweisungen können für einen Prozessor eines Universalcomputers, eines spezifischen Computers oder einer anderen programmierbaren Datenverarbeitungsvorrichtung bereitgestellt werden, um eine Maschine zu produzieren, so dass die Anweisungen, die über den Prozessor des Computers oder der anderen programmierbaren Datenverarbeitungsvorrichtung ausgeführt werden, ein Mittel für das Umsetzen der in dem einen oder den mehreren Ablaufplan- und/oder Blockschaubildblöcken angegebenen Funktionen/Aktionen zu erstellen.

[0097] Diese Computerprogrammanweisungen können auch in einem computerlesbaren Medium gespeichert werden, das einen Computer, eine andere programmierbare Datenverarbeitungsvorrichtung oder andere Einheiten anweisen kann, auf eine bestimmte Weise zu arbeiten, so dass die im computerlesbaren Medium gespeicherten Anweisungen einen Herstellungsartikel produzieren, der Anweisungen beinhaltet, die die in den einen oder mehreren Ablaufplan- und/oder Blockschaubildblöcken angegebene Funktion/Aktion umsetzen.

[0098] Die Computerprogramminstruktionen können auch in einen Computer, eine andere programmierbare Datenverarbeitungsvorrichtung oder andere Geräte geladen werden, um zu bewirken, dass eine Reihe von Betriebsschritten im Computer, auf der anderen programmierbaren Vorrichtung oder auf anderen Einheiten durchgeführt wird, um ein computerausgeführtes Verfahren zu produzieren, so dass die Anweisungen, die auf dem Computer oder auf der anderen programmierbaren Vorrichtung ausgeführt werden, Verfahren zum Umsetzen der in dem einen oder den mehreren Ablaufplan- und/oder Blockschaubildblöcken angegebenen Funktionen/Aktionen bereitstellen.

[0099] Der Ablaufplan und die Blockschaubilder in den Figuren zeigen die Architektur, die Funktionalität und den Betrieb möglicher Umsetzungen von Systemen, Verfahren und Computerprogrammprodukten gemäß verschiedener Ausführungsformen der vorliegenden Erfindung. In dieser Hinsicht kann jeder Block des Ablaufplans oder der Blockschaubilder ein Modul, ein Segment oder einen Teil eines Codes darstellen, das bzw. der eine oder mehrere ausführbare Anweisungen für die Umsetzung der einen oder mehreren angegebenen logischen Funktionen aufweist. Es sei darüber hinaus angemerkt, dass die in den Blöcken ausgewiesenen Funktionen bei einigen alternativen Umsetzungen in einer anderen Reihenfolge als in den Figuren gezeigt auftreten können. Beispielsweise können zwei aufeinanderfolgen Blöcke tatsächlich im Wesentlichen gleichzeitig ausgeführt werden, oder die Blöcke können manchmal in umgekehrter Reihenfolge ausgeführt werden, je nach Funktionalität. Es sei ferner angemerkt, dass jeder Block der Blockschaubilder und/oder der Ablaufplandarstellung und Kombinationen von Blöcken in den Blockschaubildern und/oder in der Ablaufplandarstellung durch spezifische hardwarebasierte Systeme umgesetzt sein können, die die angegebenen Funktionen oder Aktionen oder Kombinationen von spezifischen Hardware- und Computeranweisungen durchführen.

[0100] Die Systeme und Methodiken der vorliegenden Offenbarung können in einem Computersystem durchgeführt oder ausgeführt werden, das eine Verarbeitungseinheit beinhaltet, die eine oder mehrere Prozessoren und/oder Kerne, Speicher- oder andere Systemkomponenten (in der Zeichnung nicht ausdrücklich gezeigt) aufweist, die ein Computerverarbeitungssystem oder einen Computer umsetzen, das bzw. der ein Computerprogrammprodukt ausführen kann. Das Computerprogrammprodukt kann Medien aufweisen, beispielsweise eine Festplatte, ein Kompaktspeichermedium wie eine Compact Disk oder andere Speichereinheiten, die von der Verarbeitungseinheit mithilfe von dem Fachmann bekannten oder künftig bekannt werdenden Verfahren für die Bereitstellung des Computerprogrammprodukts an das Verarbeitungssystem zu Ausführungszwecken gelesen werden können.

[0101] Das Computerprogrammprodukt kann alle jeweiligen Funktionen aufweisen, die die Umsetzung der hier beschriebenen Methodiken ermöglichen, und kann die Verfahren, wenn es in ein Computersystem geladen ist, ausführen. Unter Computerprogramm, Software-Programm, Programm oder Software versteht sich

im vorliegenden Kontext jeder Ausdruck eines Satzes von Anweisungen in einer beliebigen Sprache, einem beliebigen Code oder einer beliebigen Schreibweise, der bewirken soll, dass ein System mit einer Informationsverarbeitungsfunktion eine bestimmte Funktion entweder direkt oder nach einem oder beiden des Folgenden durchführt: (a) Umwandlung in eine andere Sprache, einen anderen Code oder eine andere Schreibweise; und/oder (b) Reproduktion in einer anderen materiellen Form.

[0102] Das Computerverarbeitungssystem, das das System und das Verfahren der vorliegenden Offenbarung ausführt, kann auch eine Anzeigeeinheit, beispielsweise einen Monitor oder einen Anzeigebildschirm für das Darstellen von Ausgabeanzeigen und für das Bereitstellen einer Anzeige beinhalten, über die der Benutzer Daten eingeben und mit dem Verarbeitungssystem interagieren kann, beispielsweise im Kombination mit Eingabeeinheiten wie Tastatur und Mauseinheit oder Zeigeeinheit. Das Computerverarbeitungssystem kann auch direkt oder mittels Remote-Verbindung mit einer oder mehreren Peripherieeinheiten wie Drucker, Scanner, Lautsprecher und andere Einheiten verbunden oder gekoppelt sein. Das Computerverarbeitungssystem kann über eine oder mehrere von lokalem Ethernet, WAN-Verbindung, Internet usw. oder über andere Vernetzungsmethodiken, die unterschiedliche Rechensysteme verbinden und einen Datenaustausch zwischen diesen ermöglichen, mit einem oder mehreren anderen Verarbeitungssystemen, beispielsweise einem Server, einem anderen Remote-Computerverarbeitungssystem oder Netzwerkspeichereinheiten, verbunden oder gekoppelt sein. Die verschiedenen Funktionalitäten und Module der Systeme und Verfahren der vorliegenden Offenbarung können auf unterschiedlichen Verarbeitungssystemen verteilt oder auf einer einzelnen Plattform umgesetzt sein oder ausgeführt werden, die beispielsweise auf lokal oder im Netzwerk verteilt gespeicherte Daten zugreift.

[0103] Die hier verwendete Terminologie dient lediglich zum Beschreiben bestimmter Ausführungsformen und soll die Erfindung nicht einschränken. Wie hier verwendet, sollen die Singularformen von Artikeln wie „ein“ und „der“ auch die Pluralformen mit einschließen, außer wenn der Kontext es eindeutig anders vorgibt. Es sei ferner verstanden, dass die Ausdrücke „aufweisen“ und/oder „aufweisend“, wie in dieser Schrift verwendet, das Vorhandensein von angegebenen Merkmalen, ganzen Zahlen, Schritten, Operationen, Elementen und/oder Komponenten festlegen, das Vorhandensein oder das Hinzufügen von einem/r oder mehreren anderen Merkmalen, ganzen Zahlen, Schritten, Operationen, Elementen, Komponenten und/oder Gruppen davon jedoch nicht ausschließen.

[0104] Die entsprechenden Strukturen, Materialien, Aktionen und sämtliche Mittel oder Schritt-plus-Funktion-Elemente, falls vorhanden, in den folgenden Ansprüchen sollen jedwede Struktur, jedwedes Material oder jedwede Aktion für das Durchführen der Funktion in Kombination mit anderen beanspruchten Elementen, wie spezifisch beansprucht, beinhalten. Die Beschreibung der vorliegenden Erfindung ist zum Zwecke der Veranschaulichung und Beschreibung dargeboten, soll jedoch nicht als ausschöpfend oder die Erfindung in der offenbarten Form einschränkend verstanden werden. Für den Fachmann sind viele Änderungen und Variationen ersichtlich, ohne sich vom Umfang und Geist der Erfindung zu entfernen. Die Ausführungsform wurde gewählt und beschrieben, um die Grundsätze der Erfindung und die praktische Anwendung bestmöglich zu erläutern und um anderen Fachleuten zu ermöglichen, die Erfindung in verschiedenen Ausführungsformen mit verschiedenen Änderungen, wie sie sich für die bestimmte angedachte Verwendung eignen, zu verstehen.

[0105] Verschiedene Aspekte der vorliegenden Offenbarung können als Programm, Software oder Computeranweisungen umgesetzt sein, die in einem computer- oder maschinenlesbaren oder -nutzbaren Medium umgesetzt sind, das den Computer oder die Maschine veranlasst, die Schritte des Verfahrens durchzuführen, wenn es auf dem Computer, dem Prozessor und/oder der Maschine ausgeführt wird. Ferner wird eine von einer Maschine lesbare Programmspeichereinheit bereitgestellt, die konkret als Programm mit Anweisungen umgesetzt ist, das von der Maschine ausführbar ist, um diverse in der vorliegenden Offenbarung beschriebene Funktionalitäten und Verfahren durchzuführen.

[0106] Das System und das Verfahren der vorliegenden Offenbarung können auf einem Universalcomputer oder einem speziellen Computersystem umgesetzt sein und ausgeführt werden. Das Computersystem kann ein beliebiger Typ von bekannten oder bekannt werdenden Systemen sein und kann für gewöhnlich einen Prozessor, eine Arbeitsspeichereinheit, eine Speichereinheit, Eingabe-/Ausgabeeinheiten, interne Busse und/oder eine Datenübertragungsschnittstelle zum Austausch von Daten mit anderen Computersystemen in Verbindung mit Datenübertragungs-Hardware und -Software usw. beinhalten.

[0107] Die Ausdrücke „Computersystem“ und „Computernetzwerk“, wie sie bei der vorliegenden Anmeldung verwendet werden können, können eine Vielzahl von Kombinationen aus ortsfester/n und/oder tragbarer/n

Computer-Hardware, Software, Peripheriegeräten und Speichereinheiten beinhalten. Das Computersystem kann eine Vielzahl von einzelnen Komponenten beinhalten, die zur Zusammenarbeit vernetzt oder anderweitig verbunden sind, oder kann eine oder mehrere eigenständige Komponenten beinhalten. Die Hardware- und Software-Komponenten des Computersystems der vorliegenden Anmeldung können ortsfeste und tragbare Einheiten wie Desktop-PC, Laptop und/oder Server beinhalten und in diesen enthalten sein. Ein Modul kann eine Komponente einer Einheit, einer Software, eines Programms oder eines Systems sein, die eine gewisse „Funktion“ ausführt, die als Software, Hardware, Firmware, elektronischer Schaltkreis oder usw. umgesetzt sein kann.

[0108] Die oben beschriebenen Ausführungsformen sind veranschaulichende Beispiele, und die vorliegende Erfindung sollte nicht als auf diese bestimmten Ausführungsformen beschränkt ausgelegt werden. Somit kann der Fachmann diverse Änderungen und Modifikationen vornehmen, ohne sich vom Geist oder Umfang der Erfindung, wie in den beiliegenden Ansprüchen definiert, zu entfernen.

Patentansprüche

1. Verfahren zum bedarfsgesteuerten Streaming von Abbildern virtueller Maschinen, aufweisend:
Kopieren von mit einer ausgewählten virtuellen Maschine verknüpften Abbild-Metadaten von einem Speicher-Server, auf dem eine oder mehrere Abbildvorlagen gespeichert sind, die jeweils einer oder mehreren virtuellen Maschinen entsprechen, in einen lokalen Speicher eines Host-Computers, wobei der lokale Speicher des Host-Computers zunächst kein Abbild der ausgewählten virtuellen Maschine enthält, wobei die Abbild-Metadaten zunächst einen Kopfzeile-Abschnitt enthalten, der lediglich einen Verweis auf die Abbildvorlage aufweist, die der ausgewählten virtuellen Maschine entspricht;
Booten der ausgewählten virtuellen Maschine im Host-Computer unter Verwendung der kopierten Abbild--Metadaten und durch Zugriff auf die Abbildvorlage von dem Speicher-Server, wie während des Bootens erforderlich;
Ermöglichen, dass die ausgewählte virtuelle Maschine Daten aus der Abbildvorlage im Speicher-Server liest, die erforderlich sind, um das Ausführen der ausgewählten virtuellen Maschine im Host-Computer fortzusetzen, wenn die benötigten Daten nicht im lokalen Speicher des Host-Computers gespeichert sind;
Kopieren der gelesenen Daten der Abbildvorlage vom Speicher-Server in den lokalen Speicher des Host--Computers, wenn die gelesenen Daten der Abbildvorlage nicht im lokalen Speicher des Host-Computers gespeichert sind, wobei nachfolgende Lesevorgänge der gleichen Daten aus dem lokalen Speicher des Host-Computers durchgeführt werden;
Setzen eines Bits in einer Bitmap, wobei die Bitmap im Arbeitsspeicher und auf der Festplatte im Host--Computer aufbewahrt wird, und wobei ein Vorgang des Erstellens einer Kopie beim Lesen nur die In-Memory-Bitmap aktualisiert; und
Nutzen der Ressourcen-Leerlaufzeit, um mit der ausgewählten virtuellen Maschine verknüpfte Daten der Abbildvorlage aus dem Speicher-Server in den lokalen Speicher des Host-Computers vorab zu lesen.

2. Verfahren nach Anspruch 1, wobei die Abbild-Metadaten zunächst einen Verweis auf die Abbildvorlage und die Bitmap, die ein Bit einem entsprechenden Sektor der Abbildvorlage zuordnet, beinhaltet.

3. Verfahren nach Anspruch 1, wobei die Reaktionszeiten während des Abbild-Vorablesezugriffs überwacht werden, und wobei der Vorablesezugriff vorübergehend angehalten wird, wenn eine Reaktionszeit einen Grenzwert überschreitet.

4. Verfahren nach Anspruch 1, wobei die Größe der Bitmap proportional zur Größe der Abbildvorlage im Speicher-Server und nicht zur Größe des entsprechenden Abbilds ist, das im lokalen Speicher des Host--Computers gespeichert ist.

5. Verfahren nach Anspruch 1, wobei eine vollständige Kopie der Bitmap im Arbeitsspeicher aufbewahrt werden kann.

6. Verfahren nach Anspruch 1, wobei die Lese- und/oder Schreibanfragen der ausgewählten virtuellen Maschine für einen Sektor, dessen logische Blockadresse über die Größe der Abbildvorlage im Speicher--Server hinaus geht, durch Lesen und/oder Schreiben des Sektors direkt in den lokalen Speicher des Host--Computers bearbeitet wird, ohne dass die Bitmap überprüft und/oder die Bitmap aktualisiert wird.

7. Verfahren nach Anspruch 1, wobei eine oder mehrere, von der ausgewählten virtuellen Maschine ausgegebene Festplatten-Lese- und/oder Festplatten-Schreibanfragen nach abgeschlossenem Vorablese-

zugriff unter Verwendung von Abbilddaten ausgeführt werden, die direkt im lokalen Speicher des Host-Computers gespeichert sind, ohne dass die Bitmap überprüft und/oder die Bitmap aktualisiert wird.

8. Verfahren nach Anspruch 1, wobei der Vorgang des Erstellens einer Kopie beim Lesen die On-Disk--Bitmap nicht sofort aktualisiert.

9. Verfahren nach Anspruch 1, wobei der Vorgang des Erstellens einer Kopie beim Lesen nicht Teil des kritischen Pfads des Zurückgebens der Daten an die VM ist, und wobei die Daten asynchron im Hintergrund in das ODS-Abbild gespeichert werden, während die VM die Verarbeitung der aus dem Sicherungsabbild geladenen Daten fortsetzt.

10. Verfahren nach Anspruch 1, wobei ein Ausgangsstatus eines Datensektors, wenn der Datensektor in der Abbildvorlage im Speicher-Server zur Gänze mit Nullen gefüllt ist, in der im lokalen Speicher des Host--Computers gespeicherten Bitmap so eingestellt wird, als wäre der Datensektor bereits in den lokalen Speicher des Host-Computers kopiert.

11. Verfahren nach Anspruch 1, wobei das Verfahren in einem Hypervisor umgesetzt ist, wobei Funktionalitäten des Hypervisors ohne Änderungen an der ausgewählten virtuellen Maschine bereitgestellt werden.

12. Verfahren zum bedarfsgesteuerten Streaming von Abbildern virtueller Maschinen, aufweisend:
Kopieren von mit einer virtuellen Maschine verknüpften Abbild-Metadaten von einem Quell-Computer, auf dem eine der virtuellen Maschine entsprechende Abbildvorlage gespeichert ist, auf einen Ziel-Computer, wobei der Ziel-Computer die Abbildvorlage der virtuellen Maschine zunächst nicht enthält, wobei die Abbild-Metadaten zunächst einen Kopfzeile-Abschnitt enthalten, der lediglich einen Verweis auf die Abbildvorlage aufweist, die der ausgewählten virtuellen Maschine entspricht;
Booten der virtuellen Maschine im Ziel-Computer unter Verwendung der kopierten Abbild-Metadaten und durch Zugriff auf die Abbildvorlage von dem Speicher-Server, wie während des Bootens erforderlich;
Ermöglichen, dass die virtuelle Maschine im Ziel-Computer Daten der Abbildvorlage im Quell-Computer liest, die erforderlich sind, um das Ausführen der virtuellen Maschine im Ziel-Computer fortzusetzen, wenn die erforderlichen Daten der Abbildvorlage nicht im Ziel-Computer gespeichert sind;
Kopieren der gelesenen Daten der Abbildvorlage vom Quell-Computer auf den Ziel-Computer, wenn die gelesenen Daten der Abbildvorlage nicht im Ziel-Computer gespeichert sind, wobei bei nachfolgenden Lesungen der gleichen Daten die kopierten Daten im Ziel-Computer gelesen werden; und
Setzen eines Bits in einer Bitmap, wobei die Bitmap im Arbeitsspeicher und auf der Festplatte im Host--Computer aufbewahrt wird, und wobei ein Vorgang des Erstellens einer Kopie beim Lesen nur die In-Memory-Bitmap aktualisiert.

13. Verfahren nach Anspruch 12, wobei die Abbild-Metadaten zunächst einen Verweis auf die Abbildvorlage und die Bitmap, die ein Bit einem entsprechenden Sektor der Abbildvorlage zuordnet, beinhaltet.

14. Verfahren nach Anspruch 12, ferner beinhaltend das Bestimmen, ob die zur Ausführung der virtuellen Maschine erforderlichen Daten im Ziel-Computer gespeichert sind, indem das Bit in der Bitmap überprüft wird, wobei die virtuelle Maschine je nach Bit in der Bitmap Daten aus dem Quell-Computer oder dem Ziel-Computer liest.

15. Verfahren nach Anspruch 12, ferner beinhaltend das Nutzen von Ressourcen-Leerlaufzeit, um mit der virtuellen Maschine verknüpfte Daten der Abbildvorlage aus dem Quell-Computer in den Ziel-Computer vorab zu lesen.

16. Verfahren nach Anspruch 12, wobei die Schritte für eine Live-Migration der virtuellen Maschine vom Quell-Computer auf den Ziel-Computer durchgeführt werden, wobei kein separater Speicher-Server verwendet wird.

17. Nichtflüchtiges computerlesbares Speichermedium, auf dem ein Programm mit Anweisungen gespeichert ist, das von einer Maschine ausführbar ist, um ein Verfahren für das bedarfsgesteuerte Streaming von Abbildern virtueller Maschinen durchzuführen, aufweisend:
Kopieren von mit einer ausgewählten virtuellen Maschine verknüpften Abbild-Metadaten von einem Speicher-Server, auf dem eine oder mehrere Abbildvorlagen gespeichert sind, die jeweils einer oder mehreren virtuellen Maschinen entsprechen, in einen lokalen Speicher eines Host-Computers, wobei der lokale Speicher des Host-Computers zunächst keine Abbildvorlage der ausgewählten virtuellen Maschine enthält,

wobei die Abbild-Metadaten zunächst einen Kopfzeile-Abschnitt enthalten, der lediglich einen Verweis auf die Abbildvorlage aufweist, die der ausgewählten virtuellen Maschine entspricht;
Booten der ausgewählten virtuellen Maschine im Host-Computer unter Verwendung der kopierten Abbild--Metadaten und durch Zugriff auf die Abbildvorlage von dem Speicher-Server, wie während des Bootens erforderlich;
Ermöglichen, dass die ausgewählte virtuelle Maschine Daten aus der Abbildvorlage im Speicher-Server liest, die erforderlich sind, um das Ausführen der ausgewählten virtuellen Maschine im Host-Computer fortzusetzen, wenn die benötigten Daten nicht im lokalen Speicher des Host-Computers gespeichert sind;
Kopieren der gelesenen Daten der Abbildvorlage vom Speicher-Server in den lokalen Speicher des Host--Computers, wenn die gelesenen Daten der Abbildvorlage nicht im lokalen Speicher des Host-Computers gespeichert sind, wobei nachfolgende Lesevorgänge der gleichen Daten aus dem lokalen Speicher des Host-Computers durchgeführt werden; und
Setzen eines Bits in einer Bitmap, wobei die Bitmap im Arbeitsspeicher und auf der Festplatte im Host--Computer aufbewahrt wird, und wobei ein Vorgang des Erstellens einer Kopie beim Lesen nur die In-Memory-Bitmap aktualisiert.

18. Computerlesbares Speichermedium nach Anspruch 17, wobei die Abbild-Metadaten zunächst einen Verweis auf die Abbildvorlage beinhalten.

19. Computerlesbares Speichermedium nach Anspruch 18, wobei die Abbild-Metadaten ferner die Bitmap beinhalten, die ein Bit einem entsprechenden Sektor der Abbildvorlage zuordnet.

20. Computerlesbares Speichermedium nach Anspruch 17, ferner beinhaltend das Bestimmen, ob die zur Ausführung der ausgewählten virtuellen Maschine benötigten Daten der Abbildvorlage im lokalen Speicher des Host-Computers gespeichert sind, indem das Bit in der Bitmap überprüft wird, wobei die ausgewählte virtuelle Maschine je nach Bit in der Bitmap die Abbildvorlage im Speicher-Server oder die kopierte Abbildvorlage im lokalen Speicher des Host-Computers liest.

21. Computerlesbares Speichermedium nach Anspruch 17, ferner beinhaltend das Nutzen von Ressourcen-Leerlaufzeit, um mit der ausgewählten virtuellen Maschine verknüpfte Daten der Abbildvorlage aus dem Speicher-Server in den lokalen Speicher des Host-Computers vorab zu lesen.

22. System zum bedarfsgesteuerten Streaming von Abbildern virtueller Maschinen, aufweisend:
einen Ziel-Computer, der funktionsfähig ist, um mit einer virtuellen Maschine verknüpfte Abbild-Metadaten von einem Quell-Computer, auf dem eine der virtuellen Maschine entsprechende Abbildvorlage gespeichert ist, zu kopieren, wobei der Ziel-Computer die Abbildvorlage der virtuellen Maschine zunächst nicht beinhaltet, wobei die Abbild-Metadaten zunächst einen Kopfzeile-Abschnitt enthalten, der lediglich einen Verweis auf die Abbildvorlage aufweist, die der ausgewählten virtuellen Maschine entspricht; und
eine Speichereinheit, die lokal im Ziel-Computer angeschlossen ist,
wobei der Ziel-Computer darüber hinaus funktionsfähig ist, um die virtuelle Maschine im Ziel-Computer unter Verwendung der kopierten Abbild-Metadaten und durch Zugriff auf die Abbildvorlage von dem Speicher-Server, wie während des Bootens erforderlich, zu booten und es der virtuellen Maschine im Ziel-Computer zu ermöglichen, Daten der Abbildvorlage im Quell-Computer zu lesen, die erforderlich sind, um das Ausführen der virtuellen Maschine im Ziel-Computer fortzusetzen, wenn die benötigten Daten der Abbildvorlage nicht im Ziel-Computer gespeichert sind,
wobei der Ziel-Computer darüber hinaus funktionsfähig ist, um die gelesenen Daten der Abbildvorlage aus dem Quell-Computer in die lokal im Ziel-Computer angeschlossene Speichereinheit zu kopieren, wenn die gelesenen Daten der Abbildvorlage nicht im Ziel-Computer gespeichert sind, wobei nachfolgende Lesevorgänge der gleichen Daten aus der lokal im Ziel-Computer angeschlossenen Speichereinheit durchgeführt werden,
wobei der Ziel-Computer darüber hinaus zum Setzen eines Bits in einer Bitmap funktionsfähig ist, wobei die Bitmap im Arbeitsspeicher und auf der Festplatte im Host-Computer aufbewahrt wird, und wobei ein Vorgang des Erstellens einer Kopie beim Lesen nur die In-Memory-Bitmap aktualisiert.

23. System nach Anspruch 22, wobei die Abbild-Metadaten zunächst einen Verweis auf die Abbildvorlage beinhalten.

24. System nach Anspruch 23, wobei die Abbild-Metadaten ferner die Bitmap beinhalten, die ein Bit einem entsprechenden Sektor der Abbildvorlage zuordnet.

25. System nach Anspruch 22, ferner beinhaltend das Nutzen von Ressourcen-Leerlaufzeit, um mit der virtuellen Maschine verknüpfte Daten der Abbildvorlage aus dem Quell-Computer in den Ziel-Computer vorab zu lesen.

Es folgen 8 Seiten Zeichnungen

Anhängende Zeichnungen

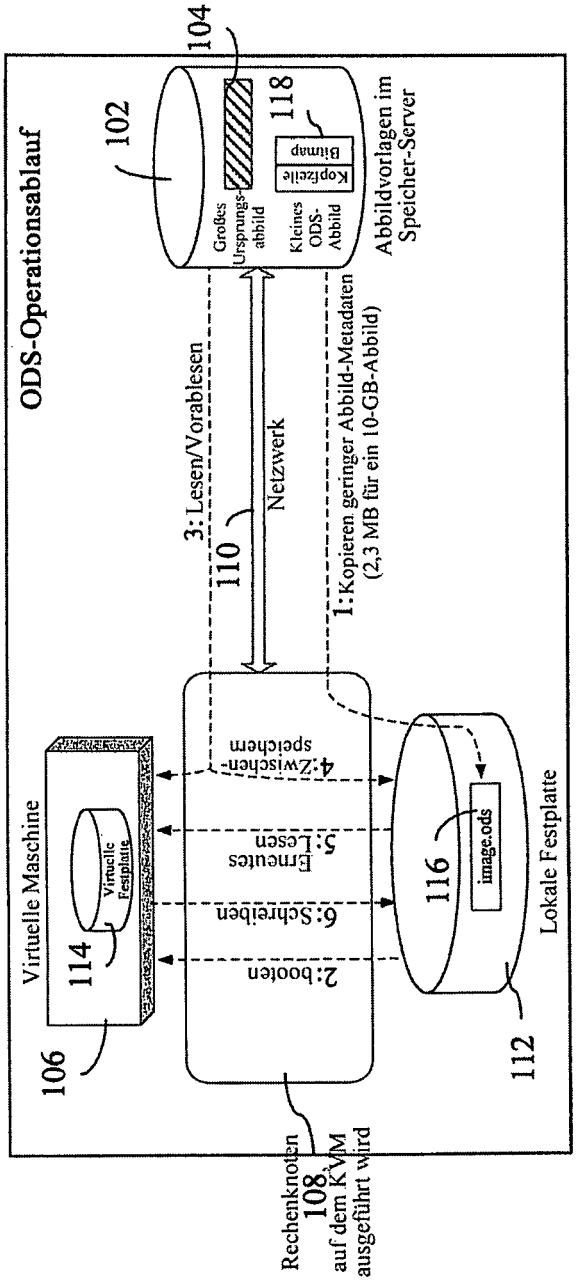
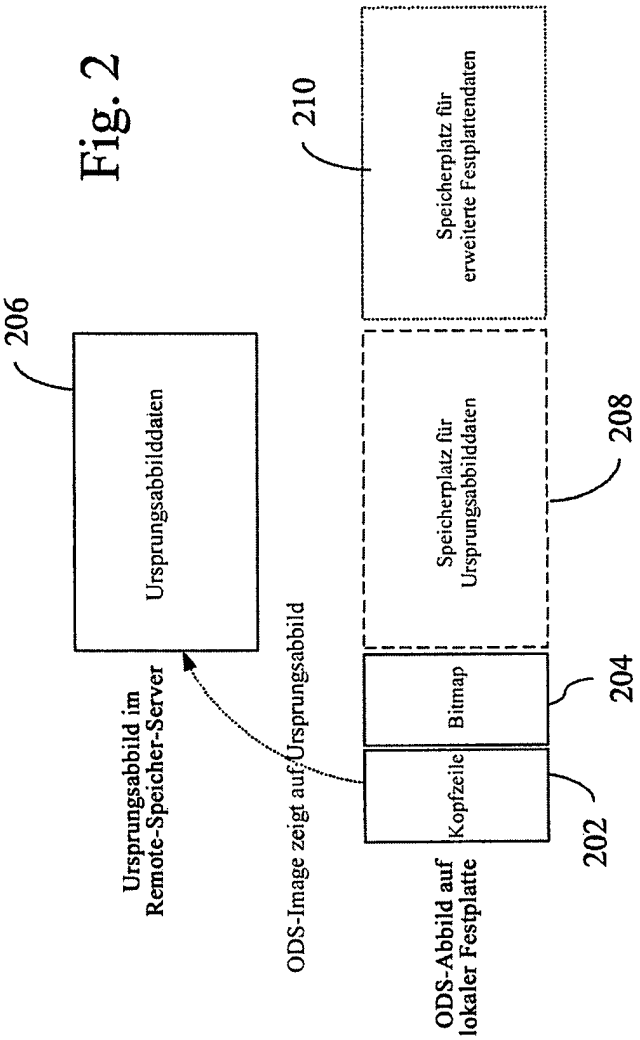


Fig. 1

Fig. 2


206
Ursprungsabbilddaten
Ursprungsabbild im Remote-Speicher-Server
ODS-Image zeigt auf Ursprungsabbild
210
Speicherplatz für erweiterte Festplattendaten
208
Speicherplatz für Ursprungsabbilddaten
204
Bitmap
202
Kopfzeile
ODS-Abbild auf lokaler Festplatte

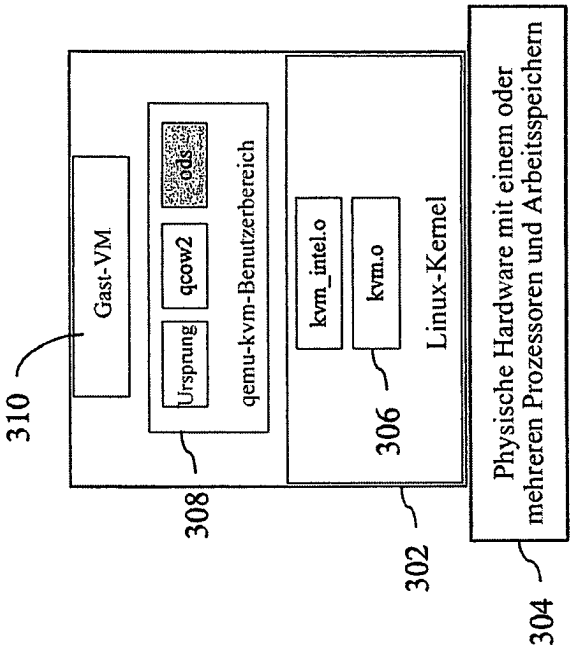
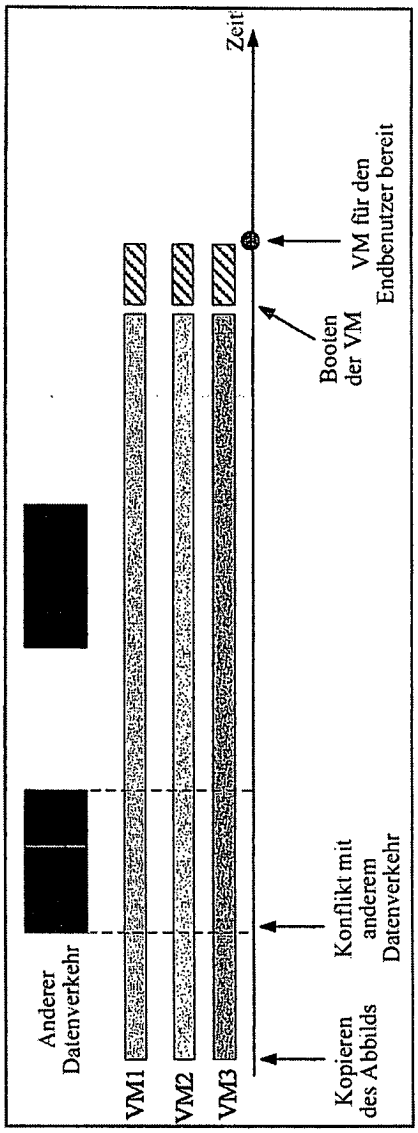
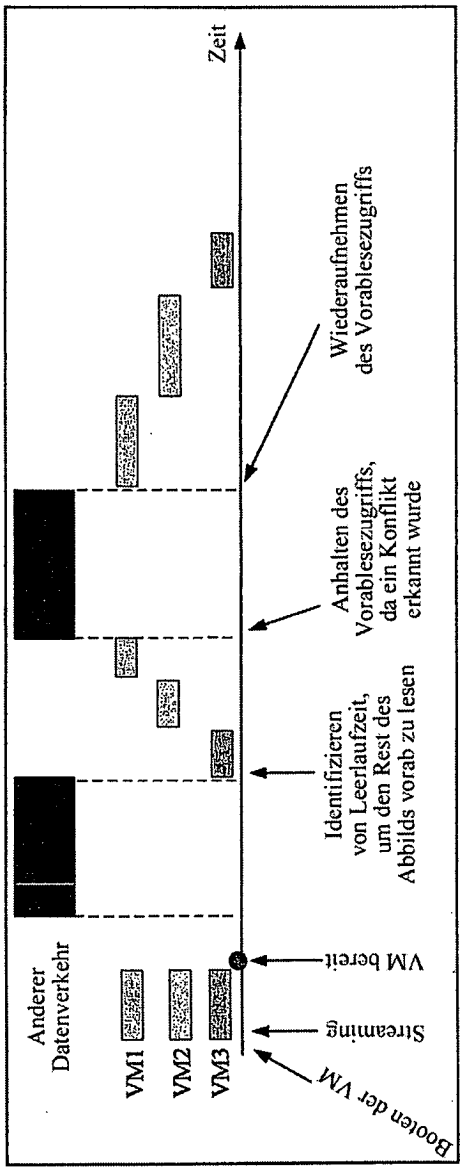

Gast-VM
Ursprung
qcow2
qemu-kvm-Benutzerbereich
kvm_intel.o
kvm.o
Linux-Kernel
Physische Hardware mit einem oder mehreren Prozessoren und Arbeitsspeichern
310
308
306
302
304

Fig. 3



Ohne ODS
(Verzögerung, bevor VM bereit ist)

Fig. 4A



Mit ODS
(schnelles Starten der VM)

Fig. 4B

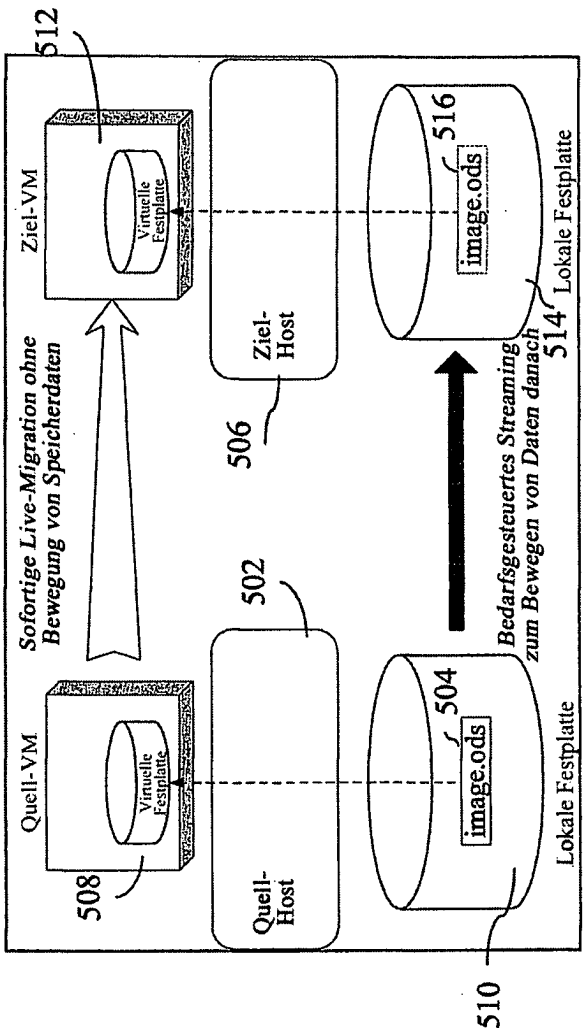

512
Ziel-VM
Virtuelle Festplatte
Sofortige Live-Migration ohne Bewegung von Speicherdaten
Quell-VM
Virtuelle Festplatte
508
Quell-Host
502
Ziel-Host
506
image.ods
504
image.ods
516
Bedarfsgesteuertes Streaming zum Bewegen von Daten danach
Lokale Festplatte
514
Lokale Festplatte
510

Fig. 5

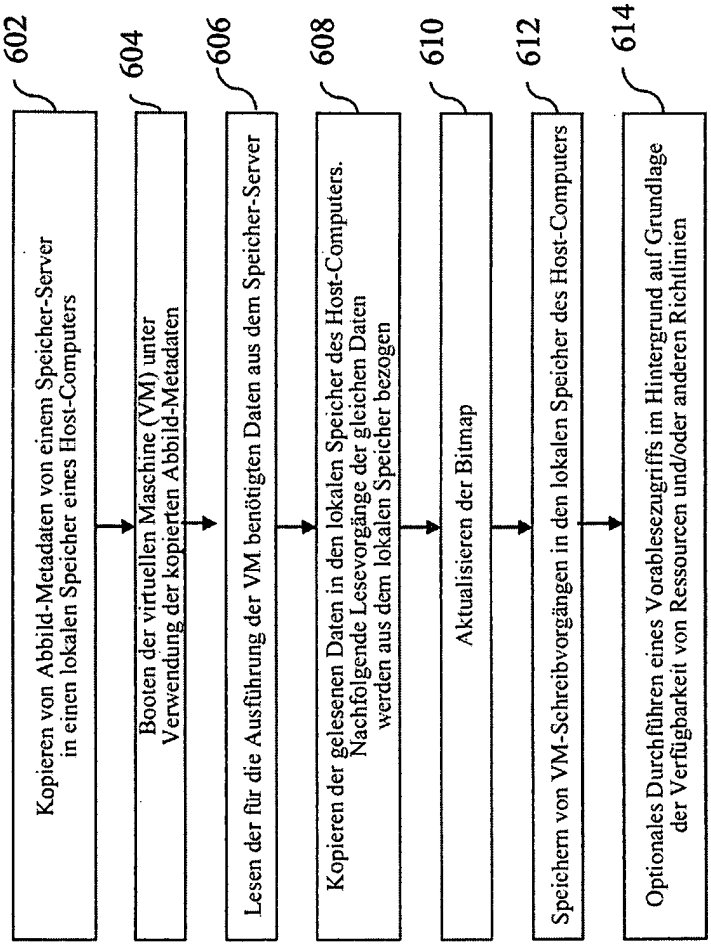

602
Kopieren von Abbild-Metadaten von einem Speicher-Server in einen lokalen Speicher eines Host-Computers
604
Booten der virtuellen Maschine (VM) unter Verwendung der kopierten Abbild-Metadaten
606
Lesen der für die Ausführung der VM benötigten Daten aus dem Speicher-Server
608
Kopieren der gelesenen Daten in den lokalen Speicher des Host-Computers. Nachfolgende Lesevorgänge der gleichen Daten werden aus dem lokalen Speicher bezogen
610
Aktualisieren der Bitmap
612
Speichern von VM-Schreibvorgängen in den lokalen Speicher des Host-Computers
614
Optionales Durchführen eines Vorablesezugriffs im Hintergrund auf Grundlage der Verfügbarkeit von Ressourcen und/oder anderen Richtlinien

Fig. 6

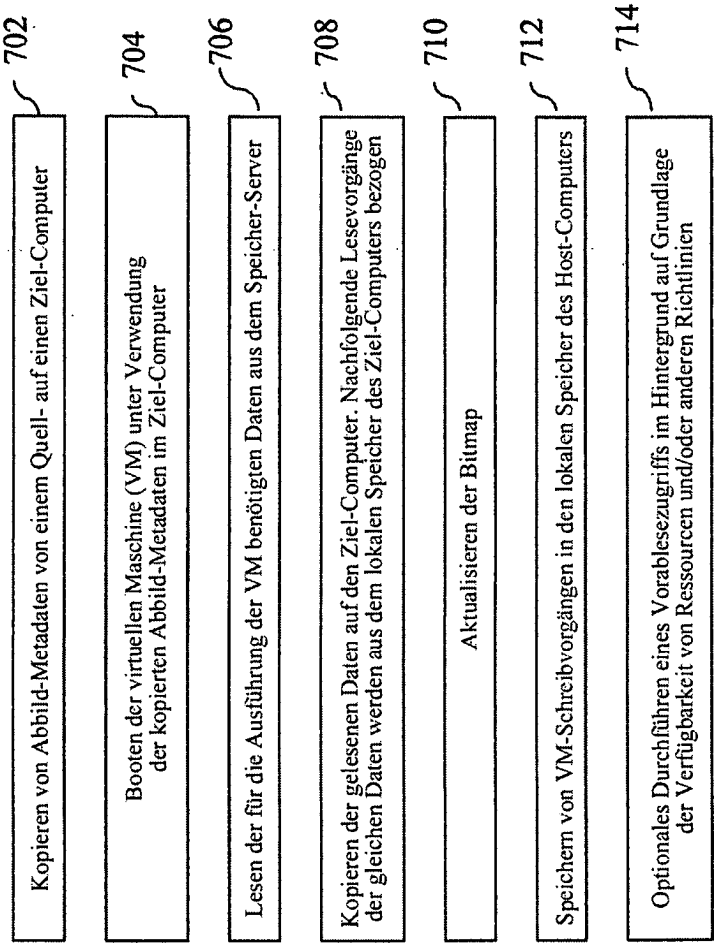

702
Kopieren von Abbild-Metadaten von einem Quell- auf einen Ziel-Computer
704
Booten der virtuellen Maschine (VM) unter Verwendung der kopierten Abbild-Metadaten im Ziel-Computer
706
Lesen der für die Ausführung der VM benötigten Daten aus dem Speicher-Server
708
Kopieren der gelesenen Daten auf den Ziel-Computer. Nachfolgende Lesevorgänge der gleichen Daten werden aus dem lokalen Speicher des Ziel-Computers bezogen
710
Aktualisieren der Bitmap
712
Speichern von VM-Schreibvorgängen in den lokalen Speicher des Host-Computers
714
Optionales Durchführen eines Vorablesezugriffs im Hintergrund auf Grundlage der Verfügbarkeit von Ressourcen und/oder anderen Richtlinien

Fig. 7

Fig. 8A

Fig. 8B

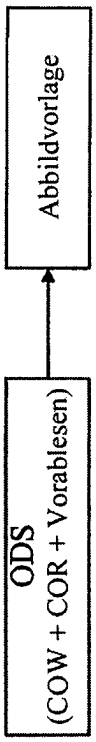


Fig. 8C

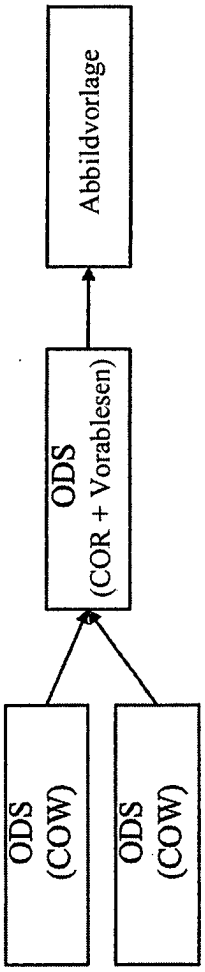


Fig. 8D

