(19) Deutsches Patent- und Markenamt

(10) **DE 11 2021 002 893 T5** 2023.03.16

(12) # Veröffentlichung

der internationalen Anmeldung mit der
(87) Veröffentlichungs-Nr.: **WO 2022/041996**
in der deutschen Übersetzung (Art. III § 8 Abs. 2 IntPatÜbkG)
(21) Deutsches Aktenzeichen: **11 2021 002 893.6**
(86) PCT-Aktenzeichen: **PCT/CN2021/102418**
(86) PCT-Anmeldetag: **25.06.2021**
(87) PCT-Veröffentlichungstag: **03.03.2022**
(43) Veröffentlichungstag der PCT Anmeldung in deutscher Übersetzung: **16.03.2023**

(51) Int Cl.: ***G06F 9/455*** (2006.01)
G06N 20/00 (2019.01)

(30) Unionspriorität:
17/000,399 **24.08.2020** **US**

(71) Anmelder:
Kyndryl, Inc., New York, NY, US

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

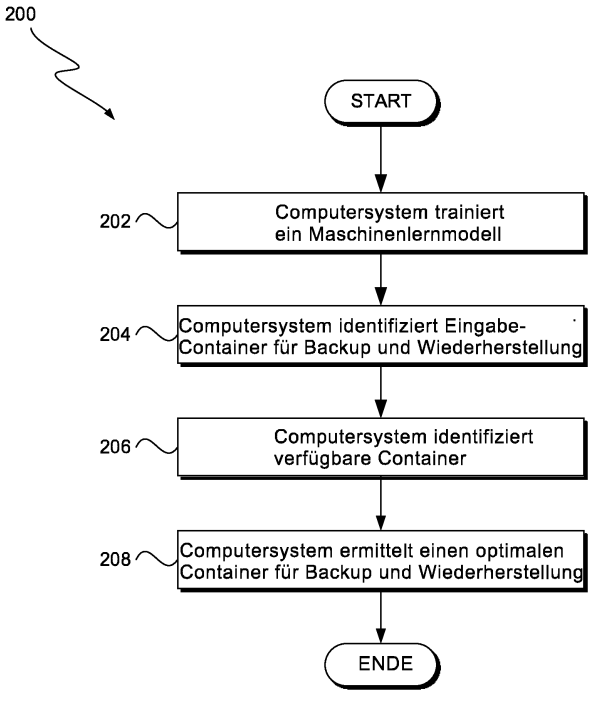
(72) Erfinder:
Rane, Girish Vilas, Beijing, CN; Khor, Pei Wen, Singapore, SG; Wong, Kim Poh, Singapore, SG

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **INTELLIGENTES BACKUP UND WIEDERHERSTELLUNG EINER CONTAINERISIERTEN UMGEBUNG**

(57) Zusammenfassung: Ein System zum Ermitteln des optimalen Backup- und Wiederherstellungsortes für Lightweight-Anwendungen wird bereitgestellt. Eine Computereinheit identifiziert einen Satz von Trainingsdaten, wobei die Trainingsdaten einen Lightweight-Container, einen entsprechenden Backup-Lightweight-Container und einen Optimierungs-Score für den Lightweight-Container und den entsprechenden Backup-Lightweight-Container identifizieren. Die Datenverarbeitungseinheit trainiert ein Maschinenlernmodell unter Verwendung der identifizierten Trainingsdaten. Die Datenverarbeitungseinheit identifiziert einen neuen Lightweight-Container für Backup. Die Datenverarbeitungseinheit ermittelt einen optimalen Backup-Lightweight-Container für den neuen Lightweight-Container unter Verwendung des trainierten Maschinenlernmodells.



Beschreibung

HINTERGRUND DER ERFINDUNG

[0001] Die vorliegende Erfindung bezieht sich allgemein auf das Gebiet der Daten-Backup und -Speicherung, und insbesondere auf das Backup und die Speicherung von Software-Containern.

[0002] Im Allgemeinen werden in einer containerisierten Umgebung Daten komprimiert und gespeichert, um eine Lightweight-Anwendung zu erstellen, die in verschiedenen Betriebssystemen implementiert werden kann, ohne den Overhead der Verknüpfung eines Betriebssystems innerhalb jeder Anwendung.

KURZDARSTELLUNG

[0003] Ausführungsformen der vorliegenden Erfindung stellen ein Verfahren, System und Programmprodukt bereit, um den optimalen Backup- und Wiederherstellungsort für Lightweight-Anwendungen zu ermitteln.

[0004] Eine erste Ausführungsform umfasst ein Verfahren, wobei ein oder mehrere Prozessoren einen Satz von Trainingsdaten identifizieren, wobei die Trainingsdaten einen Lightweight-Container, einen entsprechenden Backup-Lightweight-Container, und einen Optimierungs-Score für den Lightweight-Container und den entsprechenden Backup-Lightweight-Container identifizieren. Ein oder mehrere Prozessoren trainieren ein Maschinenlernmodell unter Verwendung der identifizierten Trainingsdaten. Ein oder mehrere Prozessoren identifizieren einen neuen Lightweight-Container für Backup. Der eine oder die mehreren Prozessoren ermitteln einen optimalen Backup-Lightweight-Container für den neuen Lightweight-Container unter Verwendung des trainierten Maschinenlernmodells.

[0005] Eine zweite Ausführungsform umfasst ein Computerprogrammprodukt, das ein oder mehrere computerlesbare Speichermedien und Programmanweisungen beinhaltet, die auf dem einen oder den mehreren computerlesbaren Speichermedien gespeichert sind. Die Programmanweisungen beinhalten Programmanweisungen zum Identifizieren eines Satzes von Trainingsdaten, wobei die Trainingsdaten einen Lightweight-Container, einen entsprechenden Backup-Lightweight-Container, und einen Optimierungs-Score für den Lightweight-Container und den entsprechenden Backup-Lightweight-Container identifizieren. Die Programmanweisungen beinhalten Programmanweisungen zum Trainieren eines Maschinenlernmodells unter Verwendung der identifizierten Trainingsdaten. Die Programmanweisungen beinhalten Programmanweisungen zum Identifizieren eines neuen Lightweight-Containers für Backup. Die Programmanweisungen beinhalten Programmanweisungen zum Ermitteln eines optimalen Backup-Lightweight-Containers für den neuen Lightweight-Container unter Verwendung des trainierten Maschinenlernmodells.

[0006] Eine dritte Ausführungsform umfasst ein Computersystem, das einen oder mehrere Computer-Prozessoren, ein oder mehrere computerlesbare Speichermedien und Programmanweisungen beinhaltet, die auf den computerlesbaren Speichermedien zur Ausführung durch mindestens einen der einen oder mehreren Prozessoren gespeichert sind. Die Programmanweisungen beinhalten Programmanweisungen zum Identifizieren eines Satzes von Trainingsdaten, wobei die Trainingsdaten einen Lightweight-Container, einen entsprechenden Backup-Lightweight-Container, und einen Optimierungs-Score für den Lightweight-Container und den entsprechenden Backup-Lightweight-Container identifizieren. Die Programmanweisungen beinhalten Programmanweisungen zum Trainieren eines Maschinenlernmodells unter Verwendung der identifizierten Trainingsdaten. Die Programmanweisungen beinhalten Programmanweisungen zum Identifizieren eines neuen Lightweight-Containers für Backup. Die Programmanweisungen beinhalten Programmanweisungen zum Ermitteln eines optimalen Backup-Lightweight-Containers für den neuen Lightweight-Container unter Verwendung des trainierten Maschinenlernmodells.

Figurenliste

Fig. 1 ist ein funktionales Blockschaltbild, das eine Datenverarbeitungsumgebung veranschaulicht, in der eine Computereinheit den optimalen Backup- und Wiederherstellungsort gemäß einer beispielhaften Ausführungsform der vorliegenden Erfindung ermittelt.

Fig. 2 veranschaulicht Betriebsabläufe bei der Ausführung eines Systems zum Ermitteln des optimalen Backup- und Wiederherstellungsorts auf einer Datenverarbeitungseinheit in der Umgebung von **Fig. 1**, gemäß einer beispielhaften Ausführungsform der vorliegenden Erfindung.

Fig. 3 stellt ein Komponentendiagramm gemäß mindestens einer Ausführungsform der vorliegenden Erfindung dar.

Fig. 4 stellt ein Blockschaltbild der containerisierten Umgebung gemäß mindestens einer Ausführungsform der vorliegenden Erfindung dar.

Fig. 5 stellt eine Cloud-Computing-Umgebung gemäß mindestens einer Ausführungsform der vorliegenden Erfindung dar.

Fig. 6 stellt Abstraktionsmodellschichten gemäß mindestens einer Ausführungsform der vorliegenden Erfindung dar.

Fig. 7 stellt ein Blockschaltbild von Komponenten einer oder mehrerer Datenverarbeitungseinheiten innerhalb der in **Fig. 1** dargestellten Datenverarbeitungsumgebung gemäß einer beispielhaften Ausführungsform der vorliegenden Erfindung dar.

AUSFÜHRLICHE BESCHREIBUNG

[0007] Detaillierte Ausführungsformen der vorliegenden Erfindung werden hierin unter Bezugnahme auf die begleitenden Zeichnungen offenbart. Es sollte klar sein, dass die offenbarten Ausführungsformen für potenzielle Ausführungsformen der vorliegenden Erfindung rein veranschaulichend sind und verschiedene Formen annehmen können. Außerdem ist jedes der in Verbindung mit den verschiedenen Ausführungsformen angegebene Beispiel veranschaulichend, aber nicht einschränkend zu betrachten. Ferner sind die Figuren nicht notwendigerweise maßstabsgetreu, einige Merkmale können übertrieben dargestellt sein, um Details bestimmter Komponenten zu zeigen. Daher sind hierin offenbarte spezifische strukturelle und funktionale Details nicht als einschränkend zu interpretieren, sondern als rein repräsentative Grundlage, um einen Fachmann zu lehren, die vorliegende Erfindung auf verschiedene Art zu verwenden.

[0008] Verweise in der Patentschrift auf „eine Ausführungsform“, „irgendeine Ausführungsform“, eine „beispielhafte Ausführungsform“ usw. geben an, dass die beschriebene Ausführungsform ein bestimmtes Merkmal, eine bestimmte Struktur oder Eigenschaft enthalten kann, dass aber nicht jede Ausführungsform notwendigerweise das bestimmte Merkmal, die bestimmte Struktur oder Eigenschaft enthalten muss. Darüber hinaus verweisen derartige Formulierungen nicht notwendigerweise auf dieselbe Ausführungsform. Wenn ferner ein bestimmtes Merkmal, eine bestimmte Struktur oder Eigenschaft in Verbindung mit einer Ausführungsform beschrieben wird, wird davon ausgegangen, dass es dem Wissensstand eines Fachmanns entspricht, derartige Merkmale, Strukturen oder Eigenschaften in Verbindung mit anderen Ausführungsformen einzuordnen, ungeachtet dessen, ob sie explizit beschrieben wurden.

[0009] Die vorliegende Erfindung wird nun unter Bezugnahme auf die Figuren im Detail beschrieben.

[0010] **Fig. 1** ist ein funktionales Blockschaltbild, das die Datenverarbeitungsumgebung, die allgemein als 100 bezeichnet wird, gemäß einer Ausführungsform der vorliegenden Erfindung veranschaulicht. Die Datenverarbeitungsumgebung 100 beinhaltet ein Computersystem 120, eine Client-Einheit 130 und ein Storage Area Network (SAN) 140, das über ein Netzwerk 110 verbunden ist. Das Computersystem 120 beinhaltet ein intelligentes Containerisierungs-Programm 122 und eine Computerschnittstelle 124. Die Client-Einheit 130 beinhaltet die Client-Anwendung 132 und die Client-Schnittstelle 134. Das Storage Area Network (SAN) 140 beinhaltet eine Serveranwendung 142 und eine Datenbank 144.

[0011] In verschiedenen Ausführungsformen der vorliegenden Erfindung ist das Computersystem 120 eine Datenverarbeitungseinheit, die eine eigenständige Einheit, ein Server, ein Laptop-Computer, ein Tablet-Computer, ein Netbook-Computer, ein Personal Computer (PC), ein Personal Digital Assistant (PDA), ein Desktop-Computer oder eine beliebige programmierbare elektronische Einheit sein kann, die in der Lage ist, Daten zu empfangen, zu senden und zu verarbeiten. Im Allgemeinen stellt das Computersystem 120 eine beliebige programmierbare elektronische Einheit oder eine Kombination programmierbarer elektronischer Einheiten dar, die in der Lage sind, maschinenlesbare Programmanweisungen auszuführen und Daten mit verschiedenen anderen Computersystemen (nicht dargestellt) auszutauschen. In einer anderen Ausführungsform stellt das Computersystem 120 ein Datenverarbeitungssystem dar, das geclusterte Computer und Komponenten einsetzt, um als ein einziger Bestand an nahtlosen Ressourcen zu fungieren. Im Allgemeinen kann das Computersystem 120 eine beliebige Computereinheit oder eine Kombination von Einheiten mit Zugriff auf verschiedene andere Computersysteme (nicht gezeigt) sein und ist in der Lage, das intelligente Containerisierungs-Programm 122 und die Computerschnittstelle 124 auszuführen. Das Computersystem

120 kann interne und externe Hardwarekomponenten beinhalten, wie in Bezug auf **Fig. 1** ausführlich beschrieben.

[0012] In dieser beispielhaften Ausführungsform sind das intelligente Containerisierungs-Programm 122 und die Computerschnittstelle 124 auf dem Computersystem 120 gespeichert. In anderen Ausführungsformen sind das intelligente Containerisierungs-Programm 122 und die Computerschnittstelle 124 jedoch extern gespeichert und der Zugriff auf sie erfolgt über ein Datenübertragungs-Netzwerk, z. B. das Netzwerk 110. Bei dem Netzwerk 110 kann es sich beispielsweise um ein lokales Netzwerk (LAN), ein Weitverkehrsnetzwerk (WAN) wie das Internet oder eine Kombination aus beidem handeln, und es kann eine drahtgebundene, drahtlose, faseroptische oder jede andere in der Technik bekannte Verbindung beinhalten. Im Allgemeinen kann das Netzwerk 110 eine beliebige Kombination von Verbindungen und Protokollen sein, die die Datenübertragung zwischen dem Computersystem 120, der Client-Einheit 130, dem SAN 140 und verschiedenen anderen Computersystemen (nicht dargestellt) gemäß einer gewünschten Ausführungsform der vorliegenden Erfindung unterstützen wird.

[0013] In verschiedenen Ausführungsformen der vorliegenden Erfindung können die verschiedenen anderen Computersysteme (nicht gezeigt) eine eigenständige Einheit, ein Server, ein Laptop-Computer, ein Tablet-Computer, ein Netbook-Computer, ein Personal Computer (PC), ein Desktop-Computer oder eine beliebige programmierbare elektronische Einheit sein, die in der Lage ist, Daten zu empfangen, zu senden und zu verarbeiten. In einer anderen Ausführungsform stellt das andere Computersystem ein Datenverarbeitungssystem dar, das geclusterte Computer und Komponenten einsetzt, um als ein einziger Bestand an nahtlosen Ressourcen zu fungieren. Im Allgemeinen kann es sich bei den verschiedenen anderen Computersystemen um jede beliebige Datenverarbeitungseinheit oder Kombination von Einheiten mit Zugriff auf das System 120, die Client-Einheit 130, das SAN 140 und das Netzwerk 110 handeln, die in der Lage ist, das intelligente Containerisierungs-Programm 122 und die Computerschnittstelle 124 auszuführen. Die verschiedenen anderen Computersysteme können interne und externe Hardwarekomponenten beinhalten, wie in Bezug auf **Fig. 1** dargestellt und ausführlich beschrieben.

[0014] In der in **Fig. 1** dargestellten Ausführungsform hat das intelligente Containerisierungs-Programm 122 zumindest teilweise Zugriff auf die Client-Anwendung 132 und kann Daten, die auf dem Computersystem 120 gespeichert sind, mit der Client-Einheit 130, dem SAN 140 und verschiedenen anderen Computersystemen (nicht dargestellt) austauschen. Insbesondere definiert das intelligente Containerisierungs-Programm 122 einen Benutzer des Computersystems 120, der Zugriff auf Daten hat, die auf der Client-Einheit 130 und der Datenbank 144 gespeichert sind.

[0015] Das intelligente Containerisierungs-Programm 122 ist der veranschaulichenden Einfachheit halber in **Fig. 1** dargestellt. In verschiedenen Ausführungsformen der vorliegenden Erfindung stellt das intelligente Containerisierungs-Programm 122 logische Operationen dar, die auf dem Computersystem 120 ausgeführt werden, wobei die Computerschnittstelle 124 die Möglichkeit bietet, diese logischen Operationen, die gemäß dem intelligenten Containerisierungs-Programm 122 verwaltet und ausgeführt werden, einzusehen. In einigen Ausführungsformen stellt das intelligente Containerisierungs-Programm 122 ein kognitives KI-System dar, das Eingaben und Ausgaben (E/A) verarbeitet und analysiert. Außerdem arbeitet das intelligente Analyseprogramm 122, wenn es eine kognitive KI-Verarbeitung ausführt, so, dass es aus den analysierten E/A lernt und den optimalen Backup- und Speicherort für die auf der Lightweight-Anwendung gespeicherten Daten ermittelt.

[0016] Das Computersystem 120 beinhaltet die Computerschnittstelle 124. Die Computerschnittstelle 124 stellt eine Schnittstelle zwischen dem Computersystem 120, der Client-Einheit 130 und dem SAN 140 bereit. In einigen Ausführungsformen kann die Computerschnittstelle 124 eine grafische Benutzerschnittstelle (graphical user interface, GUI) oder eine Web-Benutzerschnittstelle (web user interface, WUI) sein und kann Text, Dokumente, Webbrowser, Fenster, Benutzeroptionen, Anwendungsschnittstellen und Anweisungen für die Operation anzeigen und beinhaltet die Informationen (z. B. Grafik, Text und Ton), die das Programm einem Benutzer präsentiert, sowie Steuerungssequenzen, die der Benutzer zur Steuerung des Programms einsetzt. In einigen Ausführungsformen greift das Computersystem 120 auf Daten zu, die von der Client-Einheit 130 und/oder dem SAN 140 über eine clientbasierte Anwendung übertragen werden, die auf dem Computersystem 120 läuft. Zum Beispiel beinhaltet das Computersystem 120 eine mobile Anwendungssoftware, die eine Schnittstelle zwischen dem Computersystem 120, der Client-Einheit 130 und dem SAN 140 bereitstellt.

[0017] In verschiedenen Ausführungsformen ist die Client-Einheit 130 eine Datenverarbeitungseinheit, die eine eigenständige Einheit, ein Server, ein Laptop, ein Tablet-Computer oder eine beliebige programmierbare elektronische Einheit sein kann, die in der Lage ist, Daten zu empfangen, zu senden und zu verarbeiten. Im Allgemeinen stellt das Computersystem 120 eine beliebige programmierbare elektronische Einheit oder eine Kombination programmierbarer elektronischer Einheiten dar, die in der Lage sind, maschinenlesbare Programmanweisungen auszuführen und Datenübertragungen mit verschiedenen anderen Computersystemen (nicht dargestellt) durchzuführen.

[0018] Die Client-Anwendung 132 ist in **Fig. 1** der veranschaulichenden Einfachheit halber dargestellt. In verschiedenen Ausführungsformen der vorliegenden Erfindung stellt die Client-Anwendung 132 logische Operationen dar, die auf der Client-Einheit 130 ausgeführt werden, wobei die Client-Schnittstelle 134 die Möglichkeit verwaltet, diese logischen Operationen einzusehen, die gemäß der Client-Anwendung 132 verwaltet und ausgeführt werden. In verschiedenen Ausführungsformen definiert die Client-Anwendung 132 einen Benutzer der Client-Einheit 130, der Zugriff auf Daten hat, die auf dem Computersystem 120 und/oder der Datenbank 142 gespeichert sind.

[0019] Das Storage Area Network (SAN) 140 ist ein Speichersystem, das die Serveranwendung 142 und die Datenbank 144 beinhaltet, das SAN 140 kann eine oder mehrere Computereinheiten, Server, Server-Cluster, Webserver, Datenbanken und Speichergeräte beinhalten, ist aber nicht darauf beschränkt. Das SAN 140 dient dem Datenaustausch mit dem Computersystem 120, der Client-Einheit 130 und verschiedenen anderen Computereinheiten (nicht dargestellt) über ein Netzwerk, z. B. das Netzwerk 110. Zum Beispiel kommuniziert SAN 140 mit dem intelligenten Containerisierungs-Programm 122, um Daten zwischen dem Computersystem 120, der Client-Einheit 130 und verschiedenen anderen Computereinheiten (nicht dargestellt) zu übertragen, die mit dem Netzwerk 110 verbunden sind. Das SAN 140 kann eine beliebige Datenverarbeitungseinheit oder eine Kombination von Einheiten sein, die in kommunikativer Verbindung mit einem lokalen IoT-Netzwerk stehen, d. h. einem Netzwerk, das verschiedene Datenverarbeitungseinheiten beinhaltet, einschließlich, aber nicht beschränkt auf ein Computersystem 120 und eine Client-Einheit 130, um die hierin beschriebene Funktionalität bereitzustellen. Das SAN 140 kann interne und externe Hardwarekomponenten beinhalten, wie in Bezug auf **Fig. 6** beschrieben. Die vorliegende Erfindung erkennt an, dass **Fig. 1** eine beliebige Anzahl von Datenverarbeitungseinheiten, Servern, Datenbanken und/oder Speichergeräten beinhalten kann, und die vorliegende Erfindung ist nicht lediglich auf das in **Fig. 1** Dargestellte beschränkt. So sind in einigen Ausführungsformen einige oder alle Merkmale und Funktionen des SAN 140 als Teil des Computersystems 120, der Client-Einheit 130 und/oder einer anderen Datenverarbeitungseinheit enthalten. In ähnlicher Weise sind in einigen Ausführungsformen einige der Merkmale des Computersystems 120 als Teil des SAN 140 und/oder einer anderen Datenverarbeitungseinheit enthalten.

[0020] Außerdem stellen in einigen Ausführungsformen das Computersystem 120 und das SAN 140 eine Datenverarbeitungs-Plattform dar oder sind Teil davon. Cloud-Computing ist ein Modell oder eine Dienstbereitstellung zur Ermöglichung eines mühelosen, bedarfsweisen Zugriffs über das Netzwerk auf einen gemeinsam genutzten Bestand an konfigurierbaren Datenverarbeitungsressourcen (z. B. Netzwerken, Netzwerkbandbreite, Servern, Rechenleistung, Hauptspeicher, Plattenspeicher, Anwendungen, virtuellen Maschinen und Diensten), die schnell bereitgestellt und mit äußerst geringem Verwaltungsaufwand oder Dialogverkehr mit einem Anbieter des Dienstes freigegeben werden können. Ein Cloud-Modell kann Merkmale wie Selbstbedienung nach Bedarf, Umfassender Netzwerkzugriff, Gemeinsame Nutzung physischer Ressourcen, Unverzügliche Anpassbarkeit an den Ressourcenbedarf und Messung der Service-Nutzung, kann durch Dienst-Modelle wie ein Platform as a Service (PaaS)-Modell, ein Infrastructure as a Service (IaaS)-Modell und ein Software as a Service (SaaS)-Modell dargestellt werden und kann als verschiedene Einsatzmodelle implementiert werden, darunter eine private Cloud, eine Gemeinschafts-Cloud, eine öffentliche Cloud und eine hybride Cloud.

[0021] Das SAN 140 ist in **Fig. 1** der veranschaulichenden Einfachheit halber dargestellt. Es versteht sich jedoch, dass SAN 140 gemäß verschiedener Ausführungsformen eine beliebige Anzahl von Datenbanken beinhalten kann, die gemäß der Funktionalität der Serveranwendung 142 verwaltet werden. Im Allgemeinen stellt die Datenbank 144 Daten dar und die Serveranwendung 142 stellt einen Code dar, der die Fähigkeit bereitstellt, bestimmte Aktionen in Bezug auf eine andere physische oder virtuelle Ressource durchzuführen, sowie die Fähigkeit verwaltet, die Daten zu verwenden und zu ändern. In einer alternativen Ausführungsform kann das intelligente Containerisierungs-Programm 122 auch eine beliebige Kombination der vorgenannten Merkmale darstellen, bei der die Serveranwendung 142 Zugriff auf die Datenbank 144 hat. Um verschiedene Aspekte der vorliegenden Erfindung zu veranschaulichen, werden Beispiele von Serveranwendungen 142

dargestellt, in denen das intelligente Containerisierungs-Programm 122 ein oder mehrere lokale IoT-Netzwerke darstellt, aber nicht darauf beschränkt ist.

[0022] In einigen Ausführungsformen sind die Serveranwendung 142 und die Datenbank 144 auf dem SAN 140 gespeichert. In verschiedenen Ausführungsformen können die Serveranwendung 142 und die Datenbank 144 jedoch auch extern gespeichert werden und über ein Datenübertragungsnetzwerk, wie das Netzwerk 110, wie oben beschrieben, im Zugriff sein.

[0023] Das intelligente Containerisierungs-Programm 122 ist in **Fig. 1** der veranschaulichenden Einfachheit halber dargestellt, das Computersystem 120 kann jedoch eine beliebige Anzahl von Logiken und/oder Programmen beinhalten, die gemäß dem intelligenten Containerisierungs-Programm 122 verwaltet werden. Im Allgemeinen verwaltet das Computersystem 120 den Zugriff auf das intelligente Containerisierungs-Programm 122, das eine physische oder virtuelle Ressource darstellt. In einigen Ausführungsformen beinhaltet das intelligente Containerisierungs-Programm 122 bestimmte Informationen und Codes, die, wenn sie ausgeführt werden, dem Computersystem 120 ermöglichen, spezifische Maßnahmen in Bezug auf eine andere physische und/oder virtuelle Ressource zu ergreifen, basierend auf, aber nicht beschränkt auf, diese bestimmten Informationen und Codes. Mit anderen Worten, in solchen Ausführungsformen verwaltet das intelligente Containerisierungs-Programm 122 zumindest die Fähigkeit des Computersystems 120, verschiedene Aktionen in Bezug auf eine oder mehrere physische Ressourcen und/oder virtuelle Ressourcen durchzuführen. In einigen Ausführungsformen kann das intelligente Containerisierungs-Programm 122 auch eine beliebige Kombination der vorgenannten Elemente verkörpern. Zur Veranschaulichung verschiedener Aspekte der vorliegenden Erfindung werden Beispiele für intelligente Containerisierungs-Programme 122 dargestellt, in denen das intelligente Containerisierungs-Programm 122 eines oder mehrere der Folgenden beinhaltet, jedoch nicht darauf beschränkt ist: eine Benutzerprofil-Anforderung, eine Backup- und Wiederherstellungs-Anforderung und ein intelligentes Lernen zur Optimierung des Backup- und Wiederherstellungsortes, wobei die veranschaulichenden Ausführungsformen des intelligenten Containerisierungs-Programms 122 jedoch nicht darauf beschränkt sind. Ausführungsformen der vorliegenden Erfindung erkennen an, dass das intelligente Containerisierungs-Programm 122 andere Formen von Backup- und Wiederherstellungsanforderungen beinhalten kann, die im Stand der Technik bekannt sind.

[0024] In verschiedenen Ausführungsformen der vorliegenden Erfindung stellt das intelligente Containerisierungs-Programm 122 ein Maschinenlernmodell dar, wobei das Maschinenlernmodell Eingaben/Ausgaben (E/A) umfasst. In verschiedenen Ausführungsformen analysiert das intelligente Containerisierungs-Programm 122 E/A und erlernt ein Muster von vorhergesagten Ergebnissen auf Grundlage der Ausgabe. In verschiedenen Ausführungsformen analysiert das intelligente Containerisierungs-Programm 122 Eingabedaten und ermittelt ein vorhergesagtes Muster von Ergebnissen auf Grundlage des historisch erlernten Musters von vorhergesagten Ergebnissen, ist aber nicht darauf beschränkt.

[0025] In verschiedenen Ausführungsformen der vorliegenden Erfindung müssen die Lightweight-Anwendungen von Zeit zu Zeit als Back-up gesichert und wiederhergestellt werden. Die vorliegende Erfindung sieht vor, dass die Knoten und die Speichereinheiten mit einem API-Server verbunden sind, der direkt mit der Cloud verbunden ist. Außerdem sind zusätzliche Knoten in der Lage, hinzugeschaltet zu werden, um auf zusätzlichen Ressourcen oder in einer virtualisierten Umgebung zu operieren, und sind in der Lage, die Lightweight-Anwendung auszuführen.

[0026] In verschiedenen Ausführungsformen der vorliegenden Erfindung erzeugt ein Benutzer der Client-Einheit 130 unter Verwendung der Client-Anwendung 132 eine Softwareanwendung und erzeugt (i) ein Manifest, (ii) ein Image (z.B., das den Container beschreibt), und (iii) den aktuellen Container. In einigen Ausführungsformen erzeugt die Client-Anwendung 132 einen Lightweight-Container, der die Daten der Software-Anwendung enthält, und überträgt den Lightweight-Container zur Speicherung an das SAN 140.

[0027] In verschiedenen Ausführungsformen empfängt das SAN 140 den Lightweight-Container und speichert unter Verwendung der Serveranwendung 142 den Lightweight-Container auf der Datenbank 144. Es sollte gewürdigt werden, dass die Datenbank 144 der veranschaulichenden Einfachheit halber dargestellt ist. In verschiedenen Ausführungsformen stellt die Datenbank 144 eine Ansammlung von Datenbanken, Servern, Speichereinheiten usw. dar, die alle miteinander verbunden sind, um Daten zu teilen und zu übertragen.

[0028] In verschiedenen Ausführungsformen der vorliegenden Erfindung erzeugt das intelligente Containerisierungs-Programm 122 eine Anforderung zur Backup- und Wiederherstellung. In verschiedenen Ausführungsformen identifiziert das intelligente Containerisierungs-Programm 122 mindestens einen Master-Knoten

und mindestens einen Worker-Knoten, der mit mindestens einer ersten Speichereinheit (z.B. Datenbank 144) verbunden ist. Außerdem identifiziert das intelligente Containerisierungs-Programm 122 eine oder mehrere Speichereinheiten, wobei an jede zusätzliche Speichereinheit mindestens eine Mehrzahl von Worker-Nodes angeschlossen ist. Die vorliegende Erfindung erkennt, dass der Master-Knoten und die Worker-Knoten für die Ausführung der containerisierten Anwendungen zuständig sind. In verschiedenen Ausführungsformen führen der Master-Knoten und die Worker-Knoten die Lightweight-Anwendung aus.

[0029] In verschiedenen Ausführungsformen der vorliegenden Erfindung arbeitet das intelligente Containerisierungs-Programm 122, um den optimierten Backup-Ort zu identifizieren, um die Lightweight-Anwendungen vor Datenverlust oder Korruption wiederherzustellen. In verschiedenen Ausführungsformen identifiziert das intelligente Containerisierungs-Programm 122 einen oder mehrere Backup-Ort(e) und konstruiert die optimale Herangehensweise für das Backup der Daten und die Wiederherstellung der Daten aus der containerisierten Umgebung. Außerdem identifiziert das intelligente Containerisierungs-Programm 122 den optimalen Speicherort für die Daten in den Lightweight-Containern, um die Daten mit einer Schwellengeschwindigkeit für eine optimale Wiederherstellung der Daten abzurufen. Schließlich nutzt das Maschinenlernmodell der Ausführungsformen der vorliegenden Erfindung die historischen Daten zur Identifizierung des optimalen Datenspeichers, um den optimalen Backup- und Wiederherstellungsort zu erreichen, basierend auf, aber nicht beschränkt auf, die intelligente Algorithmusformel.

[0030] Ausführungsformen der vorliegenden Erfindung erkennen, dass ein Container-Image eine statische Datei ist, die ausführbaren Programmcode für den auszuführenden Container enthält. In verschiedenen Ausführungsformen erzeugt das intelligente Containerisierungs-Programm 122 ein neues Container-Image, wenn Änderungen an den Bibliotheken und dem Code für den gegebenen Container vorgenommen werden. Das intelligente Containerisierungs-Programm 122 schützt das Container-Image, indem es das Image auf einem Repository speichert, um es vor Katastrophen und Zerstörung des Containers selbst zu schützen. In verschiedenen Ausführungsformen verwendet das intelligente Containerisierungs-Programm 122 die Backup- und Wiederherstellungsalgorithmen, um die Container-Images während einer Katastrophe oder einer Zerstörung des Containers wiederherzustellen.

[0031] Ausführungsformen der vorliegenden Erfindung erkennen, dass Container häufig Daten erstellen, die die Lebensdauer des Containers überdauern, und um diese Aufgabe zu bewältigen, mountet das intelligente Containerisierungs-Programm 122 ein Volume über ein Netzwerk-Dateisystem (NFS) in einem Objektspeicher oder einem ähnlichen Mechanismus und schreibt die Daten auf dieses Volume. In einigen Ausführungsformen stellt das intelligente Containerisierungs-Programm 122 eine Verbindung zu einer Datenbank her, um Daten zu speichern, auf die externe Benutzer zugreifen können. In verschiedenen Ausführungsformen verwendet das intelligente Containerisierungs-Programm 122 den Backup- und Wiederherstellungsalgorithmus, um die Identifizierung einer zum Backup bestimmten Umgebung während einer kritischen Situation des Containers (z. B. bei einer Katastrophe oder Zerstörung) bereitzustellen und sicherzustellen, dass die Daten ohne Datenverlust wiederhergestellt werden. In verschiedenen Ausführungsformen hostet das intelligente Containerisierungs-Programm 122 die Daten in einer Cloud-Umgebung, um große Transaktionen zu unterstützen.

[0032] In verschiedenen Ausführungsformen der vorliegenden Erfindung werden rohe und unstrukturierte Daten durch das intelligente Containerisierungs-Programm 122 identifiziert. Das intelligente Containerisierungs-Programm 122 verwendet den intelligenten Backup- und Wiederherstellungs-Algorithmus für (i) Ort-Abruf nach dem optimalen Verfahren, (ii) das Latenzverfahren, (iii) Zeiger- und Nicht-Zeiger-Daten und (iv) die spezifische zu verwendende Backup-Software. Zumindest auf Grundlage des intelligenten Backup- und Wiederherstellungsalgorithmus ermittelt das intelligente Containerisierungs-Programm 122 den optimalen Backup-Ort.

[0033] In verschiedenen Ausführungsformen definieren der Ort und die Konstruktion der Backup-Daten die optimalste Herangehensweise an den Backup- und Wiederherstellungsalgorithmus, um die auf der Lightweight-Anwendung gespeicherten Daten wiederherzustellen. In verschiedenen Ausführungsformen identifiziert das intelligente Containerisierungs-Programm 122 den besten Speicherort, der den schnellsten Abruf der Daten von den Master-Knoten und den Worker-Knoten ermöglicht, wobei dieser Schwellenwert für die Geschwindigkeit des Abrufs den optimalen Speicherort für die Wiederherstellung der Daten definiert. In verschiedenen Ausführungsformen wird innerhalb des Backup- und Wiederherstellungsverfahrens intelligentes Lernen verwendet, um sicherzustellen, dass der optimale Speicherort zumindest auf Grundlage des Backup- und Wiederherstellungsalgorithmus (z. B. Gleichung (1)) ermittelt wird. In verschiedenen Ausführungsformen ermittelt das intelligente Containerisierungs-Programm 122 die am besten geeignete Konstruktion, nachdem es die verschiedenen Herangehensweisen für den optimalen Backup- und Wiederherstellungsort auf Grund-

lage verschiedener szenariospezifischer Beschränkungen ermittelt hat, ohne darauf beschränkt zu sein. In einer Ausführungsform und einem Beispiel ruft das intelligente Containerisierungs-Programm 122 eine einzelne Datei aus einer Mehrzahl von Dateien (z. B. Tausende von Dateien) ab, wobei das intelligente Containerisierungs-Programm 122 das Ergebnis des Backup- und Wiederherstellungsalgorithmus verwaltet, um den optimalen Speicherort für diese einzelne Datei zu berechnen.

[0034] **Fig. 2** ist ein Ablaufplan 200, der die Operationen des intelligenten Containerisierungs-Programms 122 in der Datenverarbeitungsumgebung 100 gemäß einer veranschaulichenden Ausführungsform der vorliegenden Erfindung darstellt. **Fig. 2** stellt auch bestimmte Interaktionen zwischen dem intelligenten Containerisierungs-Programm 122 und der Serveranwendung 142 dar. In einigen Ausführungsformen integrieren die in **Fig. 2** dargestellten Operationen die Ausgabe bestimmter logischer Operationen des intelligenten Containerisierungs-Programms 122, das auf dem Computersystem 120 ausgeführt wird. Es sollte gewürdigt werden, dass **Fig. 2** eine Veranschaulichung einer Ausführungsform bereitstellt und keine Einschränkungen bezüglich der Umgebungen impliziert, in denen andere Ausführungsformen implementiert werden können. Es können viele Änderungen an der dargestellten Umgebung vorgenommen werden. In einer Ausführungsform kann die Abfolge der Operationen in **Fig. 2** in beliebiger Reihenfolge durchgeführt werden. In einer anderen Ausführungsform kann die in **Fig. 2** dargestellte Abfolge von Operationen gleichzeitig durchgeführt werden. Außerdem kann die in **Fig. 2** dargestellte Reihe von Operationen bei jeder Operation abgebrochen werden. Außerdem können die in **Fig. 2** dargestellten Operationen zu jedem beliebigen Zeitpunkt wiederaufgenommen werden.

[0035] In Operation 202 trainiert das intelligente Containerisierungs-Programm 122 ein Maschinenlernmodell. In verschiedenen Ausführungsformen analysiert das intelligente Containerisierungs-Programm 122 Eingabedaten von Containern, die (i) unstrukturierte Containerdaten und (ii) Konfigurations-Daten beinhalten, aber nicht darauf beschränkt sind. Ferner analysiert das intelligente Containerisierungs-Programm 122 in verschiedenen Ausführungsformen (iii) Backup-Daten bezüglich des Backup-Containers (z. B. eines entsprechenden Backup-Lightweight-Containers). Außerdem trainiert das intelligente Containerisierungs-Programm 122 ein Maschinenlernmodell unter Verwendung des aus dem Backup- und Wiederherstellungsalgorithmus (Gleichung (1)) berechneten Scores für die Optimierung. In verschiedenen Ausführungsformen erlernt das Maschinenlernmodell ein Muster von vorhergesagten Speicherorten für die Backup- und Wiederherstellung von Daten. In verschiedenen Ausführungsformen ermittelt das intelligente Containerisierungs-Programm 122 die optimalen Speicherorte, die den höchsten Wert des intelligenten Backup- und Wiederherstellungsalgorithmus (z. B. Optimierungs-Score) darstellen. Außerdem ermittelt das intelligente Containerisierungs-Programm 122 in verschiedenen Ausführungsformen einen Schwellenwert für die Häufigkeit der Vorkommnisse, bei denen der ein oder mehrere Speicherorte häufig als optimaler Speicherort für Backup- und Wiederherstellung der Daten der Lightweight-Anwendungen berechnet werden.

[0036] In verschiedenen Ausführungsformen stellt das intelligente Containerisierungs-Programm 122 dem Maschinenlernmodell einen Satz von Trainingsdaten bereit, um das Maschinenlernmodell zu trainieren, einen optimalen Speicherort für Backup und Wiederherstellung der Daten, die auf den Lightweight-Containern (z.B. Eingabe-Containern) gespeichert sind, vorherzusagen . In verschiedenen Ausführungsformen stellt das intelligente Containerisierungs-Programm 122 dem Maschinenlernmodell Daten bereit, die die Konfigurations-Daten und die unstrukturierten Daten der Lightweight-Anwendungen sowie die Backup-Speicherort-Daten darstellen. Außerdem berechnet das intelligente Containerisierungs-Programm 122 den Optimierungs-Score auf Grundlage mindestens des Backup- und Wiederherstellungsalgorithmus (z.B. Gleichung (1)), wobei das intelligente Containerisierungs-Programm 122 Werte des K-Means-Schwellenwerts, der Latenzbeschränkung und der Konfigurations-Daten verwendet. Das intelligente Containerisierungs-Programm 122 stellt den Optimierungs-Score für das Maschinenlernmodell bereit, um das Maschinenlernmodell zu trainieren, den optimalen Speicherort vorherzusagen. In verschiedenen Ausführungsformen wird das Maschinenlernmodell auf Grundlage zumindest der bereitgestellten Daten trainiert, wie oben beschrieben, und ist in der Lage, einen neuen Optimierungs-Score zu erzeugen und den optimalen Backup-Speicherort auf Grundlage der bereitgestellten Daten vorherzusagen, die die Konfigurations-Daten und die unstrukturierten Daten der Lightweight-Anwendungen sowie die Backup-Speicherort-Daten beinhalten, aber nicht darauf beschränkt sind.

[0037] Ausführungsformen der vorliegenden Erfindung stellen ein intelligentes Containerisierungs-Programm 122 bereit, um einen optimalen Ort für das Backup und die Wiederherstellung der Daten zu ermitteln, die auf der Lightweight-Anwendung gespeichert sind, basierend auf mindestens den unstrukturierten Daten der Lightweight-Anwendung. In verschiedenen Ausführungsformen verwendet das intelligente Containerisierungs-Programm 122 einen K-Mean-Algorithmus zur Vektorquantisierung und Partitionierung der unstruktu-

rierten Daten in n-Partitionen von K-Clustern, wobei jeder Cluster zum nächstgelegenen Mittelwert der unstrukturierten Daten gehört. In verschiedenen Ausführungsformen stellen die jeweiligen Muster aus den unstrukturierten Container-Daten des neuen Lightweight-Containers die n-Partitionen von K Clustern dar.

[0038] In verschiedenen Ausführungsformen der vorliegenden Erfindung verwendet das intelligente Containerisierungs-Programm 122 den K-Mean-Algorithmus, um den auf der Lightweight-Anwendung gespeicherten Datensatz in K-vorbestimmte unterschiedliche, nicht überlappende Untergruppen zu partitionieren, wobei jeder Datenpunkt des auf der Lightweight-Anwendung gespeicherten Datensatzes lediglich zu einer singulären Gruppe gehört. Das intelligente Containerisierungs-Programm 122 berechnet die Intra-Cluster-Datenpunkte und versucht, die Intra-Cluster-Datenpunkte so ähnlich wie möglich zu machen und gleichzeitig einen Schwellenwert für die Trennung zwischen den einzelnen Clustern zu pflegen.

[0039] In verschiedenen Ausführungsformen verwendet das intelligente Containerisierungs-Programm 122 Anmerkungen, um die Backup-Anforderung für die Daten zu ermitteln. Ausführungsformen der vorliegenden Erfindung erkennen, dass Anmerkungsdaten zur Identifizierung von Mustern der zum Backup und zur Wiederherstellung bestimmten Daten verwendet werden und die Inhalte bestätigen, die zur Überprüfung des Backup- und Wiederherstellungsalgorithmus verwendet werden. In verschiedenen Ausführungsformen ermittelt das intelligente Containerisierungs-Programm 122 die höchste Wahrscheinlichkeit, dass das Backup und die Wiederherstellung auf Grundlage einer Backup- und Wiederherstellungsregel, aber nicht darauf beschränkt, ausgeführt würde, wobei die Datensätze innerhalb der vorbestimmten Kriterien und des Zeitrahmens im Backup gesichert werden müssen, und ermittelt den Datensatz, der die Backup- und Wiederherstellungsregel erfüllt.

[0040] In verschiedenen Ausführungsformen berechnet das intelligente Containerisierungs-Programm 122 ferner eine Latenzbeschränkung der einen oder mehreren Speichereinheiten. Ausführungsformen der vorliegenden Erfindung erkennen, dass das intelligente Containerisierungs-Programm 122 ein Schwellenwertverfahren verwendet, um die Latenzbeschränkung zu berechnen (z. B. einen Wert von 0,6). In verschiedenen Ausführungsformen wird die Berechnung der Latenzbeschränkung unter Verwendung der Proxy-Anforderung für den Speicherort ermittelt. Außerdem wird die Latenzbeschränkung verwendet, um die Inhalte des Backups und die Fähigkeit zur Wiederherstellung der Daten sicherzustellen. In verschiedenen Ausführungsformen verwendet das intelligente Containerisierungs-Programm 122 die Latenzbeschränkung während einer Zerstörung, um die Daten nach der Analyse des Backup- und Wiederherstellungsalgorithmus wiederherzustellen. In verschiedenen Ausführungsformen nutzt das intelligente Containerisierungs-Programm 122 den Backup- und Wiederherstellungsalgorithmus mit einem Orchestrierungs-Tool, um das Backup und die Wiederherstellung der Daten in der containerisierten Umgebung auf Grundlage verschiedener Auswirkungen des Schwellenwertverfahrens auszuführen, ohne darauf beschränkt zu sein. Außerdem verwendet das intelligente Containerisierungs-Programm 122 die Latenzbeschränkung, um zusätzliche Verfahren zur Wiederherstellung der Daten in containerisierten Umgebungen während Vorgängen der Datenwiederherstellung zu berücksichtigen.

[0041] In verschiedenen Ausführungsformen berechnet das intelligente Containerisierungs-Programm 122 ferner die identifizierten Konfigurationen und mountet die Konfiguration direkt in das Dateisystem des Backup-Containers. In verschiedenen Ausführungsformen beinhalten die Konfigurations-Dateien Daten, die bind-mount-Inhalte oder Umgebungs-Nutzungs-Variablen darstellen, aber nicht darauf beschränkt sind. In verschiedenen Ausführungsformen beinhalten die Konfigurations-Daten, ohne darauf beschränkt zu sein, Parameter für die Ressourcenverwaltung (z. B. CPU, Festplatte, System, vswamp usw.), Netzwerk-Parameter (z. B. Hostname, IP-Adresse, Netzfilter, Nameserver usw.) und sonstige Daten (z. B. Kennung, Version, Neustart erlauben, Vorlage, angepasste Matrix usw.). In verschiedenen Ausführungsformen identifiziert das intelligente Containerisierungs-Programm 122 das Container-Image, das zumindest eine statische Programmdatei beinhaltet, die den für die Ausführung des Containers erforderlichen ausführbaren Code darstellt. Ausführungsformen der vorliegenden Erfindung erkennen, dass die Container-Images die Ausführung eines bestimmten Containers ermöglichen. Außerdem erzeugt das intelligente Containerisierungs-Programm 122 ein neues Image für den Container, wenn das intelligente Containerisierungs-Programm 122 Änderungen an den Bibliotheken und dem Code für einen gegebenen Container vornehmen muss. In verschiedenen Ausführungsformen verwendet das intelligente Containerisierungs-Programm 122 das Verfahren zur schnellen Identifizierung (z.B. Wert 0,7), um sicherzustellen, dass das Wiederherstellungs-Fenster in der Lage ist, die Operations-Anforderungen des intelligenten Algorithmus zum Backup- und Wiederherstellen der Daten zu erfüllen oder zu übertreffen. Ausführungsformen der vorliegenden Erfindung erkennen ferner, dass das intelligente Containerisierungs-Programm 122 das Verfahren zur schnellen Identifizierung verwendet, um die Ausführung der Backup- und Wiederherstellungsanforderungen während seiner gesamten Dauer abzuschließen. Außerdem würde ein Durchschnittsfachmann verstehen, dass das Wiederherstellungsfenster eine erfor-

derliche Backup-Zeitspanne darstellt, wobei das Backup und die Wiederherstellung der auf den Lightweight-Anwendung gespeicherten Daten innerhalb einer Schwellen-Zeitspanne geschieht.

[0042] In verschiedenen Ausführungsformen verwendet das intelligente Containerisierungs-Programm 122 die Wertausgaben von (i) dem K-Mean-Algorithmus, (ii) dem Schwellenwertverfahren und (iii) dem Verfahren zur schnellen Identifizierung, um einen Wert für eine optimierte Backup- und Wiederherstellung der Daten auf den Lightweight-Anwendungen zu berechnen. In verschiedenen Ausführungsformen nutzt das intelligente Containerisierungs-Programm 122 den intelligenten Backup- und Wiederherstellungsalgorithmus mit einem Orchestrierungs-Tool, um die Backup- und Wiederherstellung der Daten auf den Lightweight-Anwendungen innerhalb einer minimalen Schwellen-Zeitspanne auszuführen.

[0043] Die folgende Gleichung (1) stellt eine Verarbeitung dar, die durchgeführt werden kann, um den optimalen Ort für das Backup und die Wiederherstellung von Daten für die Lightweight-Container zu ermitteln.

$$\text{Intelligenter Backup- und Wiederherstellungsalorithmus}$$

$$\text{Intelligenter Bachup- und Wiederherstellungsalgorithmus} = \text{Wahrscheinlichkeit von } \left[\text{K} - \text{Mean} + \text{Schwellenwertverfahren} + \text{Verfahren zur schnellen Identifizierung}\right] / 3$$

Gleichung (1): Intelligenter Backup- und Wiederherstellungsalgorithmus

[0044] In verschiedenen Ausführungsformen berechnet das intelligente Containerisierungs-Programm 122 auf Grundlage des intelligenten Backup- und Wiederherstellungsalgorithmus, jedoch nicht darauf beschränkt, den Optimierungs-Score für die Daten auf den Lightweight-Containern. In einigen Ausführungsformen ermittelt das intelligente Containerisierungs-Programm 122 den Optimierungs-Score für die Daten, wobei der mit dem intelligenten Backup- und Wiederherstellungsalgorithmus verknüpfte Wert (z.B. Backup-Optimierungs-Score) zwischen einem Wert von 1 und 0 liegt. In einigen Ausführungsformen berechnet das intelligente Containerisierungs-Programm 122 den Optimierungs-Score des intelligenten Backup- und Wiederherstellungsalgorithmus so, dass er gleich oder größer als 0,5 ist, wobei das intelligente Containerisierungs-Programm 122 ermittelt, dass dies ein optimaler Ort für Backup und Wiederherstellung der Daten auf den Lightweight-Anwendungen ist. In einigen Ausführungsformen berechnet das intelligente Containerisierungs-Programm 122 den Optimierungs-Score des intelligenten Backup- und Wiederherstellungsalgorithmus auf weniger als 0,5, wobei das intelligente Containerisierungs-Programm 122 ermittelt, dass dies kein optimaler Ort für die Backup- und Wiederherstellung der Daten auf den Lightweight-Anwendungen ist. In verschiedenen Ausführungsformen, wenn das intelligente Containerisierungs-Programm 122 ermittelt, dass dieser Ort nicht optimal für Backup und Wiederherstellung ist, basierend auf, aber nicht beschränkt auf, den Wert, der durch die Berechnung des intelligenten Backup- und Wiederherstellungsalgorithmus erzeugt wird, dann identifiziert das intelligente Containerisierungs-Programm 122 einen oder mehrere Speicherorte.

[0045] Ausführungsformen der vorliegenden Erfindung erkennen, dass das intelligente Containerisierungs-Programm 122 die berechneten Optimierungs-Scores der verschiedenen Backup-Container an das Maschinenlernmodell bereitstellt, um das Maschinenlernmodell zu trainieren. In verschiedenen Ausführungsformen lernt das Maschinenlernmodell den optimalen Speicherort für die auf den Lightweight-Anwendungen gespeicherten Daten auf Grundlage von mindestens dem berechneten Optimierungs-Score.

[0046] In Operation 204 identifiziert das intelligente Containerisierungs-Programm 122 Eingabe-Container (d.h. Lightweight-Anwendungen) für das Backup und die Wiederherstellung von Daten, die auf den Eingabe-Containern gespeichert sind. In verschiedenen Ausführungsformen identifiziert das intelligente Containerisierungs-Programm 122 Backup-Orte für die Lightweight-Anwendung. In verschiedenen Ausführungsformen identifiziert das intelligente Containerisierungs-Programm 122 einen Eingabe-Container für Backup der auf dem Eingabe-Container (z.B. Lightweight-Anwendung) gespeicherten Daten. In einigen Ausführungsformen beinhaltet der vom intelligenten Containerisierungs-Programm 122 identifizierte Eingabe-Container, ohne jedoch darauf beschränkt zu sein, unstrukturierte Containerdaten (z.B. Inhalte auf textlastiger Basis, wie z.B. Daten, Zahl, Zeitstempel, Transaktionsaufzeichnung, Containername, verschiedene Logs, usw.) und Konfigurations-Daten (z.B. Ressourcen-Management-Parameter (z.B. CPU, Festplatte, System, vswap, usw.), Netzwerk-Parameter (z.B. Hostname, IP-Adresse, Netzfilter, Nameserver, usw.) und sonstige Parameter (z.B. Kennung, Version, Neustart zulassen, Vorlage, angepasste Matrix, usw.)), die mit dem Eingabe-Container verknüpft sind.

[0047] In Operation 206 identifiziert das intelligente Containerisierungs-Programm 122 verfügbare Container (d. h. Backup-Speicherorte) für Backup und Wiederherstellung. In verschiedenen Ausführungsformen identifi-

ziert das intelligente Containerisierungs-Programm 122 Backup-Orte für den optimalen Speicherort. In verschiedenen Ausführungsformen identifiziert das intelligente Containerisierungs-Programm 122 eine oder mehrere Speichereinheiten und einen oder mehrere Knoten, die in der Lage sind, den Backup-Speicherort auszuführen. Außerdem identifiziert das intelligente Containerisierungs-Programm 122 einen oder mehrere Speicherorte auf Grundlage dessen, wo die Container-Umgebung gehostet und auf die Speicher-logische-Gerätenummer (logical unit number, LUN) montiert ist, um die Daten zu speichern, die sich auf der Lightweight-Anwendung befinden, jedoch nicht darauf beschränkt.

[0048] In verschiedenen Ausführungsformen identifiziert das intelligente Containerisierungs-Programm 122 einen Backup-Container für Backup der in der Lightweight-Anwendung gespeicherten Daten (z.B. Eingabe-Container) . In einigen Ausführungsformen beinhaltet der Backup-Container, der vom intelligenten Containerisierungs-Programm 122 identifiziert wird, aber nicht darauf beschränkt ist, unstrukturierte Container-Daten (z.B. Inhalte auf textlastiger Basis wie Daten, Zahl, Zeitstempel, Transaktionsaufzeichnung, Containername, verschiedene Logs, usw.) und Konfigurations-Daten (z.B. Ressourcen-Management-Parameter (z.B. CPU, Festplatte, System, vswap, usw.), Netzwerk-Parameter (z.B. Hostname, IP-Adresse, Netzfilter, Nameserver, usw.) und verschiedene Parameter (z.B. Kennung, Version, Neustart zulassen, Vorlage, angepasste Matrix, usw.)). In verschiedenen Ausführungsformen der vorliegenden Erfindung identifiziert das intelligente Containerisierungs-Programm 122 zusätzliche Backup-Orte, wobei die zusätzlichen Backup-Orte, ohne darauf beschränkt zu sein, Standortdaten für das trainierte Maschinenlernmodell beinhalten, um einen Optimierungs-Score zu erzeugen, um den optimalen Speicherort für die auf der Lightweight-Anwendung gespeicherten Daten vorherzusagen.

[0049] In Operation 208 ermittelt das intelligente Containerisierungs-Programm 122 einen optimalen Backup- und Speicherort für die in der Lightweight-Anwendung gespeicherten Daten auf Grundlage zumindest des trainierten Maschinenlernmodells. In verschiedenen Ausführungsformen ermittelt das trainierte Maschinenlernmodell verschiedene Paarungen zwischen den Eingabe-Containern und den Backup-Containern, die von mindestens dem intelligenten Containerisierungs-Programm 122 bereitgestellt werden. Außerdem erzeugt das trainierte Maschinenlernmodell in verschiedenen Ausführungsformen einen Optimierungs-Score auf Grundlage zumindest der Daten (z. B. unstrukturierte Daten, Konfigurations-Daten und Backup-Daten), die vom intelligenten Containerisierungs-Programm 122 für jede Paarung zwischen den Eingabe-Containern und den Backup-Containern bereitgestellt werden. In verschiedenen Ausführungsformen analysiert das intelligente Containerisierungs-Programm 122 die Optimierungs-Scores und ermittelt den optimalen Speicherort für die auf dem Eingabe-Container (z.B. Lightweight-Anwendung) gespeicherten Daten auf Grundlage mindestens des höchsten Schwellen-Pegelwerts des Optimierungs-Scores.

[0050] In verschiedenen Ausführungsformen berechnet das intelligente Containerisierungs-Programm 122 die Optimierung von unstrukturierten Daten unter Verwendung eines K-Mean-Algorithmus (z.B. Mittelwert von 0,5) . Ausführungsformen der vorliegenden Erfindung erkennen, dass der K-Mean-Algorithmus ein Verfahren zur Vektorquantisierung ist und n-Variablen in K-Cluster unterteilt, wobei jede Beobachtung zu einem Cluster mit dem nächstgelegenen Mittelwert gehört. Dies bedeutet in verschiedenen Ausführungsformen, dass das intelligente Containerisierungs-Programm 122 unstrukturierte Daten aus dem Container verwendet, die Inhalte auf textlastiger Basis wie Datum, Zahl, Zeitstempel, Transaktionsaufzeichnung, Containernamen, verschiedene Logs usw. beinhalten, aber nicht darauf beschränkt sind, um die Optimierung der unstrukturierten Daten unter Verwendung des K-Mean-Algorithmus zu berechnen.

[0051] In verschiedenen Ausführungsformen berechnet das intelligente Containerisierungs-Programm 122 die Latenzbeschränkung der einen oder mehreren Speichereinheiten. Ausführungsformen der vorliegenden Erfindung erkennen, dass das intelligente Containerisierungs-Programm 122 ein Schwellenwertverfahren verwendet, um die Latenzbeschränkung zu berechnen (z. B. einen Wert von 0,6). In verschiedenen Ausführungsformen wird die Berechnung der Latenzbeschränkung verwendet, um den Proxy-Anforderung des Speicherorts zu ermitteln. Außerdem wird die Latenzbeschränkung verwendet, um die Backup-Inhalte und die Fähigkeit zur Wiederherstellung der Daten sicherzustellen. In verschiedenen Ausführungsformen verwendet das intelligente Containerisierungs-Programm 122 die Latenzbeschränkung während einer Zerstörung, um die Daten wiederherzustellen.

[0052] In verschiedenen Ausführungsformen berechnet das intelligente Containerisierungs-Programm 122 ferner die identifizierten Konfigurationen und mountet die Konfiguration direkt in das Dateisystem des Backup-Containers. In verschiedenen Ausführungsformen beinhalten die Konfigurations-Dateien Daten, die Bind-Mount-Inhalte oder Umgebungs-Nutzungs-Variablen darstellen, aber nicht darauf beschränkt sind. In verschiedenen Ausführungsformen beinhalten die Konfigurations-Daten, ohne darauf beschränkt zu sein,

Parameter für die Ressourcenverwaltung (z. B. CPU, Festplatte, System, vswamp usw.), Netzwerk-Parameter (z. B. Hostname, IP-Adresse, Netzfilter, Nameserver usw.) und sonstige Daten (z. B. Kennung, Version, Neustart erlauben, Vorlage, angepasste Matrix usw.). In verschiedenen Ausführungsformen verwendet das intelligente Containerisierungs-Programm 122 das Verfahren zur schnellen Identifizierung (z.B. Wert von 0,7), um sicherzustellen, dass das Wiederherstellungs-Fenster in der Lage ist, die Operations-Anforderungen für Backup- und Wiederherstellung der Daten zu erfüllen oder zu übertreffen.

[0053] Ausführungsformen der vorliegenden Erfindung erkennen, dass das intelligente Containerisierungs-Programm 122 die (i) unstrukturierten Daten, (ii) die Latenzbeschränkung, (iii) einen Wert des Verfahrens zur schnellen Identifizierung und (iv) die Daten des Backup-Containers dem trainierten Maschinenlernmodell bereitstellt, um einen Optimierungs-Score zu erzeugen und einen optimalen Ort für die auf der Lightweight-Anwendung gespeicherten Daten vorherzusagen. In verschiedenen Ausführungsformen analysiert das trainierte Maschinenlernmodell die bereitgestellten Daten und erzeugt einen Optimierungs-Score (z. B. den vom Backup- und Wiederherstellungsalgorithmus berechneten Wert). In verschiedenen Ausführungsformen sagt das trainierte Maschinenlernmodell den optimalen Speicherort auf Grundlage mindestens des erzeugten Optimierungs-Scores vorher. Ausführungsformen der vorliegenden Erfindung erkennen, dass der erzeugte Optimierungs-Score mit einem Backup-Container verknüpft ist, der von einem intelligenten Containerisierungs-Programm 122 identifiziert wurde.

[0054] In verschiedenen Ausführungsformen integriert sich das intelligente Containerisierungs-Programm 122 mit einem Orchestrierungs-Tool (z.B. Serveranwendung 142), um ein Backup innerhalb einer minimalen Schwellen-Zeitspanne auszuführen. In einigen Ausführungsformen ermöglicht das Orchestrierungs-Tool dem intelligenten Containerisierungs-Programm 122, das Backup und die Wiederherstellung der Daten der Lightweight-Anwendung basierend, aber nicht darauf beschränkt, auf den Werten der intelligenten Backup- und Wiederherstellungsalgorithmen der verschiedenen Speicherorte zu koordinieren. In einigen Ausführungsformen verwendet das intelligente Containerisierungs-Programm 122 den optimalen Ort für Backup und Wiederherstellung der Daten. Alternativ identifiziert das intelligente Containerisierungs-Programm 122, dass die optimalen Speicherorte alle Ressourcen verwendet haben und nicht für Backup und Wiederherstellung verwendet werden können. Für den Fall, dass das intelligente Containerisierungs-Programm 122 den optimalen Speicherort nicht identifizieren kann, berechnet das intelligente Containerisierungs-Programm 122 den Wert für den intelligenten Backup- und Wiederherstellungsalgorithmus neu.

[0055] In einer beispielhaften Ausführungsform wird das intelligente Containerisierungs-Programm 122 unter Verwendung der folgenden Daten trainiert, mit Bezug auf Operation 202. In dieser beispielhaften Ausführungsform haben wir einen Eingabe-Container Z (z. B. Hostname: exampleZ. example. com), der mindestens 4,6 Gigabyte (GB) an Daten verwendet hat und auf den zuletzt am 1. Januar zugegriffen wurde. Außerdem beinhalten die Konfigurations-Daten innerhalb der beispielhaften Ausführungsform eine CPU, die einen Einsatzort mit mindestens 8 verfügbaren Kernen, die mit 2,5 Gigahertz (GHz) arbeiten, angibt, 8 GB Arbeitsspeicher und 1 GB vswap beinhaltet. Schließlich beinhalten in dieser beispielhaften Ausführungsform die Backup-Ort-Daten einen geografischen Ort unter Bezugnahme auf die Container. In dieser beispielhaften Ausführungsform ist der Eingabe-Container Z 20 Meilen vom Backup-Container A und 100 Meilen vom Backup-Container B entfernt. Außerdem ist in dieser beispielhaften Ausführungsform haben wir Eingabe-Container Y (z. B., exampleY. example. com), der mindestens 7,2 GB an Daten verwendet hat und auf den zuletzt am 1. Dezember zugegriffen wurde und Konfigurations-Daten aufweist, die eine CPU, die einen Einsatzort mit mindestens 4 verfügbaren Kernen, die mit 2 Gigahertz (GHz) arbeiten, angibt 8 GB Arbeitsspeicher und 1 GB vswap beinhaltet, und sich 100 Meilen vom Backup-Container A und 5 Meilen vom Backup-Container B entfernt befindet.

[0056] In dieser beispielhaften Ausführungsform identifiziert das intelligente Containerisierungs-Programm 122 die unstrukturierten Container-Daten und Konfigurations-Daten für den Eingabe-Container Z (z.B. exampleZ. example. com) und den Eingabe-Container Y (z.B. exampleY. example. com). Außerdem identifiziert das intelligente Containerisierungs-Programm 122 die Backup-Ort-Daten (wie z.B. Latenz-Informationen und Ort-Informationen) für Backup-Container A und für Backup-Container B. Schließlich identifiziert das intelligente Containerisierungs-Programm 122 die Anmerkungen für jede Kombination von Eingabe-Container und Backup-Container, wobei die Anmerkungen angeben, wie optimal jede Kombination von Eingabe-Container und Backup-Container ist. Zum Beispiel werden in dieser Ausführungsform die Anmerkungen unter Verwendung der oben beschriebenen Gleichung (1) erzeugt. In dieser Ausführungsform identifiziert das intelligente Containerisierungs-Programm 122 für den Eingabe-Container Z, dass der Backup-Container A einen Anmerkungswert von 0,8 hat und dass der Backup-Container B einen Anmerkungswert von 0,6 hat. Das intelligente Containerisierungs-Programm 122 identifiziert auch für den Eingabe-Container Y, dass der Backup-Container

A einen Anmerkungswert von 0,5 und der Backup-Container B einen Anmerkungswert von 0,6 hat. Diese Werte werden in Kombination mit den (i) unstrukturierten Container-Daten, (ii) den Konfigurations-Daten und (iii) den Backup-Ort-Daten verwendet, um das Maschinenlernmodell des intelligenten Containerisierungs-Programms 122 zu trainieren, um neue Annotationswerte (oder „Backup-Optimierungs-Scores“) für neue Kombinationen von Eingabe-Containern und Backup-Containern zu erzeugen, die dem Modell in Zukunft bereitgestellt werden.

[0057] In dieser beispielhaften Ausführungsform, unter Bezugnahme auf Operation 204, identifiziert das intelligente Containerisierungs-Programm 122 den Eingabe-Container W für Backup und Wiederherstellung. In dieser beispielhaften Ausführungsform besagen die unstrukturierten Daten für den Eingabe-Container W (z. B. exampleW. example. com), dass der Eingabe-Container W mindestens 2,2 Gigabyte (GB) an Daten verwendet hat und dass auf ihn zuletzt am 15. Januar zugegriffen wurde. Außerdem enthalten die Konfigurations-Daten für den Eingabe-Container Weine CPU, die einen Einsatzort mit mindestens 8 verfügbaren Kernen, die mit 2,3 Gigahertz (GHz) arbeiten, angibt, und 8 GB Arbeitsspeicher sowie 1 GB vswap beinhaltet. Zu beachten ist auch, dass der Eingabe-Container W 20 Meilen vom Backup-Container A und 100 Meilen vom Backup-Container B entfernt ist.

[0058] Um mit der beispielhaften Ausführungsform fortzufahren, verwendet das intelligente Containerisierungs-Programm 122 unter Bezugnahme auf Operation 206 sein trainiertes Maschinenlernmodell, um zu erzeugen: (i) einen ersten Optimierungs-Score für den Eingabe-Container W und den Backup-Ort A, und (ii) einen zweiten Optimierungs-Score für den Eingabe-Container W und den Backup-Ort B, unter Verwendung der unstrukturierten Container-Daten und Konfigurations-Daten des Eingabe-Containers W und der Backup-Ort-Daten des Backup-Ortes A bzw. des Backup-Ortes B. Insbesondere erzeugt das intelligente Containerisierungs-Programm 122 einen ersten Backup-Optimierungs-Score von 0,6 für Backup-Ort A und einen zweiten Backup-Optimierungs-Score von 0,5 für Backup-Ort B.

[0059] In dieser beispielhaften Ausführungsform analysiert das intelligente Containerisierungs-Programm 122 die Ausgabe des Maschinenlernmodells und ermittelt den optimalen Ort für den Lightweight-Container (z.B. Eingabe-Container W) auf Grundlage von mindestens den erzeugten Backup-Optimierungs-Scores. In diesem Beispiel wählt das intelligente Containerisierungs-Programm 122 den Backup-Ort A als optimalen Ort für das Backup des Eingabe-Containers W aus, basierend auf mindestens dem Backup-Ort A, der einen höheren Optimierungs-Score (0,6) aufweist als der Backup-Ort B (0,5).

[0060] **Fig. 3** ist ein Komponentendiagramm (300), das die Komponenten der containerisierten Umgebung von **Fig. 1** darstellt. In verschiedenen Ausführungsformen beinhaltet die containerisierte Umgebung mindestens einen Master-Knoten 301 und eine Mehrzahl von Worker-Knoten 302A-302C, wobei die Lightweight-Anwendung auf den Master- und Worker-Knoten gespeichert ist. Ausführungsformen der vorliegenden Erfindung stellen bereit, dass die Arbeitsknoten 302A-C der veranschaulichenden Einfachheit halber dargestellt sind und dass es eine beliebige Anzahl von Arbeitsknoten (z. B. 302C) geben kann und die vorliegende Erfindung hierauf nicht beschränkt ist. In verschiedenen Ausführungsformen sind der Master-Knoten und die Worker-Knoten kommunikativ mit den Speichereinheiten 304A und 304B zum Backup und Wiederherstellung der auf den Lightweight-Anwendungen gespeicherten Daten verbunden. Ausführungsformen der vorliegenden Erfindung stellen dar, dass Speicher 1 304A und Speicher 2 304B der veranschaulichenden Einfachheit halber dargestellt sind und dass eine beliebige Anzahl von Speichereinheiten in Ausführungsformen der vorliegenden Erfindung beinhaltet sein kann. In verschiedenen Ausführungsformen wird der Backup- und Wiederherstellungsalgorithmus 306 wie folgt dargestellt: Intelligenter Backup- und Wiederherstellungsalgorithmus = Wahrscheinlichkeit von [K-Mean + Schwellenwertverfahren + Verfahren zur schnellen Identifizierung] /3. Zum Beispiel ermittelt das intelligente Containerisierungs-Programm 122 den optimalen Backup- und Wiederherstellungs-Ort auf Grundlage von mindestens dem intelligenten Backup- und Wiederherstellungsalgorithmus. In verschiedenen Ausführungsformen verwendet das intelligente Containerisierungs-Programm 122 ein Orchestrierungs-Tool 308, um ein Backup und Wiederherstellung der auf der Lightweight-Anwendung gespeicherten Daten auszuführen, indem es mit einer virtuellen Maschine 310 kommuniziert, die eine Verbindung zu dem einen oder mehreren Speicherorten herstellt.

[0061] **Fig. 4** stellt ein Blockschaltbild 400 der containerisierten Umgebung von **Fig. 1** dar. Ausführungsformen der vorliegenden Erfindung stellen bereit, dass Container 434 im Blockschaltbild 400 angeordnet sind. In verschiedenen Ausführungsformen beginnt der Prozess des Backups und der Wiederherstellung an den Master-Knoten und Worker-Knoten 402, die die Lightweight-Anwendungen speichern, wie in **Fig. 3** angegeben. In verschiedenen Ausführungsformen identifiziert das intelligente Containerisierungs-Programm 122 die mit dem intelligenten Backup- und Wiederherstellungsalgorithmus verknüpften Eingaben 404, die (i) die Kon-

figurations-Daten 410, (ii) unstrukturierte Daten 412, (iii) Latenzbeschränkungen 414 und (iv) schnelle Identifizierungsdaten 416 (z.B. Kennung) beinhalten, aber nicht darauf beschränkt sind.

[0062] In verschiedenen Ausführungsformen verwendet das intelligente Containerisierungs-Programm 122 die oben identifizierten Eingaben, um den optimalen Speicherort für das Backup- und Wiederherstellungs-Verfahren unter Verwendung des Verfahrens mit intelligentem Algorithmus 406 (z. B. Gleichung (1)) zu berechnen . In verschiedenen Ausführungsformen wird die Wahrscheinlichkeit, ein optimales Ergebnis zu ermitteln, auf Grundlage von mindestens Gleichung (1) berechnet. In verschiedenen Ausführungsformen überprüft das intelligente Containerisierungs-Programm 122 die zugehörigen Anmerkungsdaten und führt, wenn das intelligente Containerisierungs-Programm 122 ermittelt, dass der optimale Speicherort berechnet wurde, die Backup- und Wiederherstellung-Operationen 408 aus (z.B. Ausgänge mit Orchestrierungs-Tool integrieren) . In verschiedenen Ausführungsformen berechnet das intelligente Containerisierungs-Programm 122 die Wahrscheinlichkeit, ein optimales Ergebnis 418 zu ermitteln, unter Verwendung von mindestens Gleichung (1) . Das intelligente Containerisierungs-Programm 122 ermittelt, ob die zugehörigen Anmerkungen 420 überprüft werden sollen. Wenn Nein, erfasst das intelligente Containerisierungs-Programm 122 die Aktivitäten und die Ausführung des optimierten Ergebnisses anhand der Wahrscheinlichkeit, ein optimiertes Ergebnis 422 zu ermitteln. Wenn JA, führt das intelligente Containerisierungs-Programm 122 das optimale Ergebnis 430 auf Grundlage zumindest der zugehörigen Anmerkung 420 aus. In verschiedenen Ausführungsformen verwendet das intelligente Containerisierungs-Programm 122 Tools 432, um die Ausführung abzuschließen. Das intelligente Containerisierungs-Programm 122 speichert die Daten für den Ergebnisvergleich 424 in einer nachfolgenden Schwellen-Zeitperiode. In verschiedenen Ausführungsformen analysiert das intelligente Containerisierungs-Programm 122 Daten (z. B. gespeicherte Daten für den Ergebnisvergleich 424), die aus vorangehenden Aktionen 426 abgeleitet wurden. Das intelligente Containerisierungs-Programm 122 speichert diese Daten als Artefakte 428. Wenn das intelligente Containerisierungs-Programm 122 ermittelt, dass der optimale Speicherort nicht berechnet wurde, berechnet das intelligente Containerisierungs-Programm 122 die Aktivitäten, die mit den auf der Lightweight-Anwendung gespeicherten Daten verknüpft sind, und führt das optimierte Ergebnis aus. Das intelligente Containerisierungs-Programm 122 speichert dann diese Daten für den Ergebnisvergleich.

[0063] Es versteht sich, dass diese Offenlegung zwar eine ausführliche Beschreibung des Cloud-Computing enthält, die Implementierung der hier aufgezeigten Lehren jedoch nicht auf eine Cloud-Computing-Umgebung beschränkt ist. Vielmehr können Ausführungsformen der vorliegenden Erfindung in Verbindung mit einer beliebigen anderen Art einer Datenverarbeitungsumgebung, die zum gegenwärtigen Zeitpunkt bereits bekannt ist oder zu einem späteren Zeitpunkt entwickelt werden wird, implementiert werden.

[0064] Cloud-Computing ist ein Modell zur Bereitstellung von Diensten, um einen mühelosen, bedarfsweisen Zugriff über das Netzwerk auf einen gemeinsam genutzten Bestand an konfigurierbaren Datenverarbeitungsressourcen (z. B. Netzwerken, Netzwerkbandbreite, Servern, Rechenleistung, Hauptspeicher, Plattenspeicher, Anwendungen, virtuellen Maschinen und Diensten) zu ermöglichen, die schnell bereitgestellt werden können, das Modell kann mindestens fünf Merkmale, mindestens drei Dienst-Modelle und mindestens vier Einsatzmodelle beinhalten.

[0065] Die Merkmale sind wie folgt:

> Selbstbedienung nach Bedarf: Ein Cloud-Nutzer kann einseitig Datenverarbeitungsfunktionen wie zum Beispiel Serverzeit und Netzwerkspeicher nach Bedarf automatisch bereitstellen, ohne dass eine Person mit dem Anbieter des Dienstes in Dialogverkehr treten muss.

[0066] Umfassender Netzwerkzugriff: Funktionen stehen über ein Netzwerk zur Verfügung und der Zugriff auf die Funktionen erfolgt über standardmäßige Mechanismen, welche die Nutzung durch heterogene Thin-Client- oder Thick-Client-Plattformen (z. B. Mobiltelefone, Laptops und persönliche digitale Assistenten (PDAs) unterstützen.

[0067] Gemeinsame Nutzung physischer Ressourcen: Die Datenverarbeitungsressourcen des Anbieters werden unter Verwendung eines Modells mit mehreren Mandanten (Multi-Tenant-Modell) zusammengeführt, um mehrere Nutzer zu bedienen, wobei verschiedene physische und virtuelle Ressourcen entsprechend dem Bedarf dynamisch zugewiesen und neu zugewiesen werden. Man hat ein Gefühl der Standort-Unabhängigkeit, da der Nutzer im Allgemeinen keine Kontrolle über die genauen Standorte der bereitgestellten Ressourcen hat oder den genauen Standort nicht kennt, diesen gegebenenfalls aber auf einer höheren Abstraktionsebene (z. B. Land, Staat/Bundesland oder Rechenzentrum) angeben kann.

[0068] Unverzügliche Anpassbarkeit an den Ressourcenbedarf: Funktionen können schnell und elastisch bereitgestellt werden, in manchen Fällen automatisch, um eine schnelle horizontale Skalierbarkeit zu ermöglichen, und sie können schnell freigegeben werden, um eine schnelle vertikale Skalierbarkeit zu ermöglichen. Dem Nutzer erscheinen die zur Versorgung vorhandenen Funktionen oftmals unerschöpflich und sie können jederzeit in jeder beliebigen Qualität erworben werden.

[0069] Messung der Service-Nutzung: Cloud-Systeme steuern und optimieren die Nutzung der Ressourcen automatisch, indem sie auf einer bestimmten Abstraktionsebene, die für die Art des Dienstes (z. B. Speicherkapazität, Rechenleistung, Bandbreite und aktive Benutzerkonten) geeignet ist, eine Zählerfunktion vorteilhaft einsetzen. Die Nutzung der Ressourcen kann überwacht, gesteuert und gemeldet werden, was Transparenz sowohl für den Anbieter als auch den Nutzer des in Anspruch genommenen Dienstes schafft.

[0070] Die Dienst-Modelle sind wie folgt:

Software as a Service (SaaS): Die dem Nutzer zur Verfügung gestellte Funktion besteht in der Nutzung der Anwendungen des Anbieters, die in einer Cloud-Infrastruktur ausgeführt werden. Auf die Anwendungen kann von verschiedenen Client-Einheiten über eine Thin-Client-Schnittstelle wie zum Beispiel einen Webbrowser (z. B. webbasierte eMail) zugegriffen werden. Der Nutzer verwaltet oder steuert die zugrunde liegende Cloud-Infrastruktur einschließlich des Netzwerks, der Server, der Betriebssysteme, des Speichers oder selbst einzelner Anwendungsfunktionen nicht, wobei es die mögliche Ausnahme in Form von eingeschränkten benutzerspezifischen Einstellungen bei der Konfiguration der Anwendung gibt.

[0071] Platform as a Service (PaaS): Die den Nutzern zur Verfügung gestellte Funktion besteht darin, auf der Cloud-Infrastruktur vom Nutzer erzeugte oder erworbene Anwendungen aufzusetzen, die mittels Programmiersprachen und Werkzeugen, die von dem Anbieter unterstützt werden, erzeugt wurden. Der Nutzer verwaltet oder steuert die zugrunde liegende Cloud-Infrastruktur einschließlich der Netzwerke, der Server, der Betriebssysteme oder des Speichers nicht, hat aber die Kontrolle über die eingesetzten Anwendungen und möglicherweise über Konfigurationen der Umgebung, in der Anwendungen gehostet werden (Application Hosting Environmental Configurations).

[0072] Infrastructure as a Service (IaaS): Die dem Nutzer zur Verfügung gestellte Funktion besteht in der Bereitstellung von Rechenleistung, Speicherkapazität, Netzwerken und anderen grundlegenden Datenverarbeitungsressourcen, auf denen der Kunde beliebige Software, die Betriebssysteme und Anwendungen einschließen kann, installieren und ausführen kann. Der Nutzer verwaltet oder steuert die zugrunde liegende Cloud-Infrastruktur nicht, hat aber die Kontrolle über Betriebssysteme, Speicher, eingesetzte Anwendungen und möglicherweise die eingeschränkte Kontrolle über ausgewählte Komponenten für den Netzwerkbetrieb (z. B. Zugangsschutzsysteme für den Hostrechner).

[0073] Die Einsatzmodelle sind wie folgt:

Private Cloud: Die Cloud-Infrastruktur wird ausschließlich für eine Organisation betrieben. Sie kann von der Organisation oder einem Dritten verwaltet werden und kann sich auf dem Betriebsgelände oder außerhalb des Betriebsgeländes befinden.

[0074] Gemeinschafts-Cloud: Die Cloud-Infrastruktur wird von mehreren Organisationen gemeinsam genutzt und unterstützt eine bestimmte Gemeinschaft, die gemeinsame Interessen (z. B. Aufgaben, Sicherheitsanforderungen, Richtlinien und Anforderungen an die Einhaltung der Richtlinien (Compliance-Anforderungen)) hat. Sie kann von den Organisationen oder einem Dritten verwaltet werden und kann sich auf dem Betriebsgelände oder außerhalb des Betriebsgeländes befinden.

[0075] Öffentliche Cloud: Die Cloud-Infrastruktur wird der allgemeinen Öffentlichkeit oder einer großen Industriegruppe zur Verfügung gestellt und gehört einer Organisation, die Cloud-Dienste verkauft.

[0076] Hybride Cloud: Die Cloud-Infrastruktur setzt sich aus zwei oder mehr Clouds (privaten, Gemeinschafts- oder öffentlichen Clouds) zusammen, die individuelle Entitäten bleiben, welche jedoch durch eine standardisierte oder firmeneigene Technologie miteinander verbunden sind, die die Portierbarkeit von Daten und Anwendungen ermöglicht (z. B. den kombinierten Einsatz von privaten und öffentlichen Clouds (Cloud-Bursting) für einen Lastausgleich zwischen den Clouds).

[0077] Eine Cloud-Computing-Umgebung ist dienstorientiert, wobei der Schwerpunkt auf Zustandslosigkeit, loser Kopplung, Modularität und semantischer Interoperabilität liegt. Das Kernstück des Cloud-Computing bildet eine Infrastruktur, die ein Netzwerk aus untereinander verbundenen Knoten umfasst.

[0078] Unter Bezugnahme auf **Fig. 5** ist eine veranschaulichende Cloud-Computing-Umgebung 50 dargestellt. Wie gezeigt ist, weist die Cloud-Computing-Umgebung 50 einen oder mehrere Cloud-Computing-Knoten 10 auf, mit denen vom Cloud-Nutz verwendete lokale Datenverarbeitungseinheiten wie der elektronische Assistent (PDA, Personal Digital Assistant) oder das Mobiltelefon 54A, der Desktop Computer 54B, der Laptop Computer 54C und/oder das Automobil-Computer-System 54N Daten austauschen können. Die Knoten 10 können miteinander Daten austauschen. Sie können physisch oder virtuell in ein oder mehrere Netzwerke wie private, Benutzergemeinschafts-, öffentliche oder hybride Clouds gruppiert werden (nicht gezeigt), wie vorstehend beschrieben wurde, oder in eine Kombination daraus. Dies ermöglicht es der Cloud-Computing-Umgebung 50, Infrastruktur, Plattformen und/oder Software als Dienst anzubieten, für die ein Cloud-Nutzer keine Ressourcen auf einer lokalen Datenverarbeitungseinheit vorhalten muss. Es sei darauf hingewiesen, dass die Arten von in **Fig. 5** gezeigten Datenverarbeitungseinheiten 54A bis N lediglich veranschaulichend sein sollen und dass die Datenverarbeitungsknoten 10 und die Cloud-Computing-Umgebung 50 über eine beliebige Art Netzwerk und/oder über eine beliebige Art von über ein Netzwerk aufrufbarer Verbindung (z.B. unter Verwendung eines Web-Browsers) mit einer beliebigen Art von computergestützter Einheit Daten austauschen können.

[0079] Unter Bezugnahme auf **Fig. 6** wird ein Satz von funktionalen Abstraktionsschichten gezeigt, die durch die Cloud-Computing-Umgebung 50 (**Fig. 5**) bereitgestellt werden. Es sollte von vornherein klar sein, dass die in **Fig. 6** gezeigten Komponenten, Schichten und Funktionen lediglich veranschaulichend sein sollen und Ausführungsformen der Erfindung nicht darauf beschränkt sind. Wie abgebildet ist, werden die folgenden Schichten und entsprechenden Funktionen bereitgestellt.

[0080] Eine Hardware- und Software-Schicht 60 enthält Hardware- und Software-Komponenten. Zu Beispielen für Hardware-Komponenten gehören: Mainframe Computer 61; auf der RISC-(Reduced Instructions Set Computer) Architektur beruhende Server 62; Server 63; Blade-Server 64; Speichereinheiten 65; und Netzwerke sowie Netzwerkkomponenten 66. In einigen Ausführungsformen beinhalten Software-Komponenten eine Netzwerk-Anwendungsserver-Software 67 und eine Datenbank-Software 68.

[0081] Eine Virtualisierungsschicht 70 stellt eine Abstraktionsschicht bereit, aus der die folgenden Beispiele für virtuelle Einheiten bereitgestellt werden können: virtuelle Server 71, virtueller Speicher 72, virtuelle Netzwerke 73, darunter virtuelle private Netzwerke, virtuelle Anwendungen und Betriebssysteme 74; und virtuelle Clients 75.

[0082] In einem Beispiel kann eine Verwaltungsschicht 80 die nachfolgend beschriebenen Funktionen bereitstellen. Eine Ressourcen-Bereitstellung 81 stellt die dynamische Beschaffung von Datenverarbeitungsressourcen sowie anderen Ressourcen bereit, die zum Durchführen von Aufgaben innerhalb der Cloud-Computing-Umgebung verwendet werden. Ein Messen und eine Preisfindung 82 stellen die Kostenverfolgung beim Verwenden von Ressourcen innerhalb der Cloud-Computing-Umgebung sowie die Abrechnung oder Rechnungsstellung für den Verbrauch dieser Ressourcen bereit. In einem Beispiel können diese Ressourcen Anwendungs-Software-Lizenzen aufweisen. Die Sicherheit stellt die Identitätsüberprüfung für Cloud-Nutzer und Aufgaben sowie Schutz für Daten und andere Ressourcen bereit. Ein Benutzerportal 81 stellt Nutzern und Systemadministratoren den Zugang zu der Cloud-Computing-Umgebung bereit. Eine Verwaltung des Dienstumfangs 84 stellt die Zuordnung und Verwaltung von Cloud-Computing-Ressourcen bereit, so dass die benötigten Dienstziele erreicht werden. Ein Planen und Erfüllen von Vereinbarungen zum Dienstumfang (SLA, Service Level Agreement) 85 stellt die Anordnung vorab und die Beschaffung von Cloud-Computing-Ressourcen, für die eine zukünftige Anforderung vorausgesehen wird, gemäß einem SLA bereit.

[0083] Eine Arbeitslastschicht 90 stellt Beispiele für die Funktionalität bereit, für welche die Cloud-Computing-Umgebung verwendet werden kann. Zu Beispielen für Arbeitslasten und Funktionen, die von dieser Schicht bereitgestellt werden können, gehören: Abbildung und Navigation 91; Software-Entwicklung und Lebenszyklusverwaltung 92; Bereitstellung von Ausbildung in virtuellen Klassenzimmern 93; Datenanalytikverarbeitung 94; Transaktionsverarbeitung 95; und Bereitstellung von besänftigenden Ausgaben 96.

[0084] **Fig. 7** stellt ein Blockschaltbild (700) von Komponenten des Computersystems 120, der Client-Einheit 130 und des SAN 140 gemäß einer veranschaulichenden Ausführungsform der vorliegenden Erfindung dar. Es sollte gewürdigt werden, dass **Fig. 7** lediglich eine Veranschaulichung einer Ausführungsform darstellt

und keine Einschränkungen bezüglich der Umgebungen impliziert, in denen verschiedene Ausführungsformen implementiert werden können. Es können viele Änderungen an der dargestellten Umgebung vorgenommen werden.

[0085] Das Computersystem 120, die Client-Einheit 130 und das SAN 140 beinhalten die Kommunikationsstruktur 702, die eine Datenübertragung zwischen dem/den Computer-Prozessor(en) 704, dem Arbeitsspeicher 706, dem permanenten Speicher 708, der Kommunikationseinheit 710 und der/den Eingabe/Ausgabe (E/A)-Schnittstelle(n) 712 bereitstellt. Die Kommunikationsstruktur 702 kann mit jeder Architektur implementiert werden, die geeignet ist, Datenübertragungen und/oder Steuerinformationen zwischen Prozessoren (wie z. B. Mikroprozessoren, Kommunikations- und Netzwerkprozessoren usw.), Systemarbeitsspeicher, peripheren Einheiten und allen anderen Hardwarekomponenten innerhalb eines Systems zu ermöglichen. Zum Beispiel kann die Kommunikationsstruktur 702 mit einem oder mehreren Bussen implementiert werden.

[0086] Bei dem Arbeitsspeicher 706 und dem permanenten Speicher 708 handelt es sich um computerlesbare Speichermedien. In dieser Ausführungsform beinhaltet der Arbeitsspeicher 706 einen Direktzugriffspeicher (RAM) 714 und einen Cache-Speicher 716. Im Allgemeinen kann der Arbeitsspeicher 706 jedes geeignete flüchtige oder nichtflüchtige, computerlesbare Speichermedium beinhalten.

[0087] Das intelligente Containerisierungs-Programm 122, die Computerschnittstelle 124, die Client-Anwendung 132, die Client-Schnittstelle 134, die Serveranwendung 142 und die Datenbank 144 sind in einem permanenten Speicher 708 gespeichert, um von einem oder mehreren der entsprechenden Computer-Prozessoren 704 über einen oder mehrere Speicher des Arbeitsspeichers 706 ausgeführt zu werden und/oder darauf zuzugreifen. In dieser Ausführungsform beinhaltet der permanente Speicher 708 ein magnetisches Festplattenlaufwerk. Alternativ oder außerdem zu einem magnetischen Festplattenlaufwerk kann der permanente Speicher 708 eine Festkörperfestplatte, eine Halbleiterspeichereinheit, einen Nur-Lese-Speicher (ROM), einen elektronisch löschbaren programmierbaren Nur-Lese-Speicher (EPROM), einen Flash-Speicher oder jedes andere computerlesbare Speichermedium beinhalten, das Programmanweisungen oder digitale Informationen speichern kann.

[0088] Die vom permanenten Speicher 708 verwendeten Medien können auch entfernbar sein. Zum Beispiel kann eine entfernbare Festplatte für den permanenten Speicher 708 verwendet werden. Andere Beispiele beinhalten optische und magnetische Platten, USB-Sticks und Smart Cards, die für einen Datentransfer auf ein anderes durch einen Computer lesbares Speichermedium, das ebenfalls Teil des permanenten Speichers 708 ist, in ein Laufwerk eingesetzt werden.

[0089] Die Datenübertragungseinheit 710 stellt in diesen Beispielen Datenübertragungen mit anderen Datenverarbeitungssystemen oder-einheiten bereit, einschließlich der Ressourcen des Netzwerks 110. In diesen Beispielen beinhaltet die Datenübertragungseinheit 710 eine oder mehrere Netzwerk-Schnittstellenkarten. Die Datenübertragungseinheit 710 kann Datenübertragungen unter Verwendung von physischen oder drahtlosen Datenübertragungen bereitstellen. Das intelligente Containerisierungs-Programm 122, die Computerschnittstelle 124, die Client-Anwendung 132, die Client-Schnittstelle 134, die Serveranwendung 142 und die Datenbank 144 können über die Datenübertragungseinheit 710 in den permanenten Speicher 508 heruntergeladen werden.

[0090] Die E/A-Schnittstelle(n) 712 ermöglicht bzw. ermöglichen die Eingabe und Ausgabe von Daten mit anderen Einheiten, die mit dem Computersystem 120, der Client-Einheit 130 und dem SAN 140 verbunden sein können. Beispielsweise kann die E/A-Schnittstelle 712 eine Verbindung zu externen Einheiten 718 wie einer Tastatur, einem Tastenfeld, einem Berührungsbildschirm und/oder einer anderen geeigneten Eingabeeinheit bereitstellen. Externe Einheiten 718 können auch tragbare computerlesbare Speichermedien beinhalten, wie zum Beispiel USB-Sticks, tragbare optische oder magnetische Festplatten und Speicherkarten. Software und Daten, die zur Ausführung von Ausführungsformen der vorliegenden Erfindung verwendet werden, z. B. das intelligente Containerisierungs-Programm 122, die Computerschnittstelle 124, die Client-Anwendung 132, die Client-Schnittstelle 134, die Serveranwendung 142 und die Datenbank 144, können auf solchen tragbaren, computerlesbaren Speichermedien gespeichert und über die E/A-Schnittstelle(n) 712 auf den permanenten Speicher 708 geladen werden. Die E/A-Schnittstelle(n) 712 ist bzw. sind auch mit einer Anzeige 720 verbunden.

[0091] Die Anzeige 720 stellt einen Mechanismus zur Anzeige von Daten für einen Benutzer bereit und kann beispielsweise ein Computer-Monitor oder ein Fernsehbildschirm sein.

[0092] Bei der vorliegenden Erfindung kann es sich um ein System, ein Verfahren und/oder ein Computerprogrammprodukt handeln. Das Computerprogrammprodukt kann ein durch einen Computer lesbares Speichermedium (oder -medien) enthalten, auf dem durch einen Computer lesbare Programmanweisungen gespeichert sind, um einen Prozessor dazu zu veranlassen, Aspekte der vorliegenden Erfindung auszuführen.

[0093] Bei dem durch einen Computer lesbaren Speichermedium kann es sich um eine physische Einheit handeln, die Anweisungen zur Verwendung durch eine Einheit zum Ausführen von Anweisungen beibehalten und speichern kann. Das durch einen Computer lesbare Speichermedium kann zum Beispiel eine elektronische Speichereinheit, eine magnetische Speichereinheit, eine optische Speichereinheit, eine elektromagnetische Speichereinheit, eine Halbleiter-Speichereinheit oder jede geeignete Kombination aus dem Vorgenannten sein, es ist aber nicht darauf beschränkt. Zu einer nicht erschöpfenden Liste von spezifischeren Beispielen des durch einen Computer lesbaren Speichermediums gehören die Folgenden: eine tragbare Computerdiskette, eine Festplatte, ein Direktzugriffsspeicher (RAM), ein Nur-Lese-Speicher (ROM), ein löschbarer programmierbarer Nur-Lese-Speicher (EPROM bzw. Flash-Speicher), ein statischer Direktzugriffsspeicher (SRAM), ein tragbarer CD-ROM, eine DVD, ein Speicher-Stick, eine Diskette, eine mechanisch codierte Einheit wie zum Beispiel Lochkarten oder erhabene Strukturen in einer Rille, auf denen Anweisungen gespeichert sind, und jede geeignete Kombination des Vorgenannten. Ein durch einen Computer lesbares Speichermedium soll, wie hierin verwendet, nicht als flüchtige Signale an sich aufgefasst werden, wie zum Beispiel Funkwellen oder andere sich frei ausbreitende elektromagnetische Wellen, elektromagnetische Wellen, die sich durch einen Wellenleiter oder andere Übertragungsmedien ausbreiten (z.B. durch ein Lichtwellenleiterkabel geleitete Lichtimpulse) oder durch einen Draht übertragene elektrische Signale.

[0094] Hierin beschriebene, durch einen Computer lesbare Programmanweisungen können von einem durch einen Computer lesbaren Speichermedium auf jeweilige Datenverarbeitungs-/Verarbeitungseinheiten oder über ein Netzwerk wie zum Beispiel das Internet, ein lokales Netzwerk, ein Weitverkehrsnetzwerk und/oder ein drahtloses Netzwerk auf einen externen Computer oder eine externe Speichereinheit heruntergeladen werden. Das Netzwerk kann Kupferübertragungskabel, Lichtwellenübertragungsleiter, drahtlose Übertragung, Leitwegrechner, Firewalls, Vermittlungseinheiten, Gateway-Computer und/oder Edge-Server aufweisen. Eine Netzwerkadapterkarte oder Netzwerkschnittstelle in jeder Datenverarbeitungs-/Verarbeitungseinheit empfängt durch einen Computer lesbare Programmanweisungen aus dem Netzwerk und leitet die durch einen Computer lesbaren Programmanweisungen zur Speicherung in einem durch einen Computer lesbaren Speichermedium innerhalb der entsprechenden Datenverarbeitungs-/Verarbeitungseinheit weiter.

[0095] Bei durch einen Computer lesbaren Programmanweisungen zum Ausführen von Arbeitsschritten der vorliegenden Erfindung kann es sich um Assembler-Anweisungen, ISA-Anweisungen (Instruction-Set-Architecture), Maschinenanweisungen, maschinenabhängige Anweisungen, Mikrocode, Firmware-Anweisungen, zustandssetzende Daten oder entweder Quellcode oder Objektcode handeln, die in einer beliebigen Kombination aus einer oder mehreren Programmiersprachen geschrieben sind, darunter objektorientierte Programmiersprachen wie Smalltalk, C++ o.ä. sowie herkömmliche prozedurale Programmiersprachen wie die Programmiersprache „C“ oder ähnliche Programmiersprachen. Die durch einen Computer lesbaren Programmanweisungen können vollständig auf dem Computer des Benutzers, teilweise auf dem Computer des Benutzers, als eigenständiges Software-Paket, teilweise auf dem Computer des Benutzers und teilweise auf einem entfernt angeordneten Computer oder vollständig auf dem entfernt angeordneten Computer oder Server ausgeführt werden. In dem letzteren Szenario kann der entfernt angeordnete Computer mit dem Computer des Benutzers durch jeden Typ von Netzwerk verbunden werden, darunter ein lokales Netzwerk (LAN) oder ein Weitverkehrsnetzwerk (WAN), oder die Verbindung kann zu einem externen Computer hergestellt werden (zum Beispiel über das Internet unter Verwendung eines Internet-Dienstanbieters). In einigen Ausführungsformen können elektronische Schaltungen, darunter zum Beispiel programmierbare Logikschaltungen, im Feld programmierbare Gatter-Anordnungen (FPGA, field programmable gate arrays) oder programmierbare Logikanordnungen (PLA, programmable logic arrays) die computerlesbaren Programmanweisungen ausführen, indem sie Zustandsinformationen der computerlesbaren Programmanweisungen nutzen, um die elektronischen Schaltungen zu personalisieren, um Aspekte der vorliegenden Erfindung durchzuführen.

[0096] Aspekte der vorliegenden Erfindung werden hierin unter Bezugnahme auf Veranschaulichungen von Ablaufplänen und/oder Blockschaubildern von Verfahren, Vorrichtungen (Systemen) und Computerprogrammprodukten gemäß Ausführungsformen der Erfindung beschrieben. Es sollte klar sein, dass jeder Block der Ablaufplanveranschaulichungen und/oder der Blockschaubilder und Kombinationen von Blöcken in den Ablaufplanveranschaulichungen und/oder den Blockschaubildern mittels durch einen Computer lesbare Programmanweisungen umgesetzt werden können.

[0097] Diese durch einen Computer lesbaren Programmanweisungen können für einen Prozessor eines Allzweck-Computers, eines Spezial-Computers oder eine andere programmierbare Datenverarbeitungsvorrichtung bereitgestellt werden, um eine Maschine zu erzeugen, sodass die über den Prozessor des Computers bzw. eine andere programmierbare Datenverarbeitungsvorrichtung ausgeführten Anweisungen Mittel zur Umsetzung der in dem Block bzw. den Blöcken der Ablaufpläne und/oder der Blockschaubilder angegebenen Funktionen/Schritte erstellen. Diese durch einen Computer lesbaren Programmanweisungen können auch auf einem durch einen Computer lesbaren Speichermedium gespeichert sein, das einen Computer, eine programmierbare Datenverarbeitungsvorrichtung und/oder andere Einheiten so steuern kann, dass sie auf eine bestimmte Art funktionieren, sodass das durch einen Computer lesbare Speichermedium, auf dem Anweisungen gespeichert sind, ein Herstellungsprodukt aufweist, darunter Anweisungen, die Aspekte der/des in dem Block bzw. den Blöcken des Ablaufplans und/oder der Blockschaubilder angegebenen Funktion/Schritts umsetzen.

[0098] Die durch einen Computer lesbaren Programmanweisungen können auch auf einen Computer, eine andere programmierbare Datenverarbeitungsvorrichtung oder eine andere Einheit geladen werden, um das Ausführen einer Reihe von Arbeitsschritten auf dem Computer bzw. der anderen programmierbaren Vorrichtung oder anderen Einheit zu verursachen, um einen durch einen Computer umgesetzten Prozess zu erzeugen, sodass die auf dem Computer, einer anderen programmierbaren Vorrichtung oder einer anderen Einheit ausgeführten Anweisungen die in dem Block bzw. den Blöcken der Ablaufpläne und/oder der Blockschaubilder angegebenen Funktionen/Schritte umsetzen.

[0099] Die Ablaufpläne und Blockschaubilder in den Figuren veranschaulichen die Architektur, die Funktionalität und den Betrieb möglicher Ausführungen von Systemen, Verfahren und Computerprogrammprodukten gemäß verschiedenen Ausführungsformen der vorliegenden Erfindung. In diesem Zusammenhang kann jeder Block in den Ablaufplänen oder den Blockschaubildern ein Modul, ein Segment oder einen Teil von Anweisungen darstellen, die eine oder mehrere ausführbare Anweisungen zum Umsetzen der bestimmten logischen Funktion(en) aufweisen. In einigen alternativen Umsetzungen können die in dem Block angegebenen Funktionen in einer anderen Reihenfolge als in den Figuren gezeigt auftreten. Zum Beispiel können zwei nacheinander gezeigte Blöcke tatsächlich im Wesentlichen gleichzeitig ausgeführt werden, oder die Blöcke können manchmal in der umgekehrten Reihenfolge ausgeführt werden, was von der beteiligten Funktionalität abhängt. Es ist ferner anzumerken, dass jeder Block der Blockschaubilder und/oder der Ablaufplandarstellungen sowie Kombinationen von Blöcken in den Blockschaubildern und/oder der Ablaufplandarstellung durch spezielle auf Hardware beruhende Systeme umgesetzt werden können, welche die angegebenen Funktionen oder Handlungen durchführen oder Kombinationen aus Spezial-Hardware und Computeranweisungen ausführen.

[0100] Die hierin beschriebenen Programme werden auf Grundlage der Anwendung identifiziert, für die sie in einer bestimmten Ausführungsform der Erfindung umgesetzt werden. Es sollte jedoch klar sein, dass jede bestimmte Programm-Nomenklatur hierin aus rein praktischen Gründen verwendet wird, und die Erfindung somit nicht allein auf die Verwendung in einer bestimmten identifizierten Anwendung und/oder durch eine derartige Nomenklatur umgesetzt eingeschränkt sein soll.

[0101] Es ist zu beachten, dass der Begriff (die Begriffe) wie beispielsweise „Smalltalk“ und dergleichen in verschiedenen Jurisdiktionen auf der ganzen Welt Gegenstand von Markenrechten sein kann (können) und hier lediglich unter Bezugnahme auf die Produkte oder Dienstleistungen verwendet wird, die durch die Marken ordnungsgemäß gekennzeichnet sind, sofern solche Markenrechte vorhanden sein können.

Patentansprüche

1. Computerimplementiertes Verfahren, wobei das Verfahren umfasst:

Identifizieren, durch einen oder mehrere Prozessoren, eines Satzes von Trainingsdaten, wobei die Trainingsdaten einen Lightweight-Container, einen entsprechenden Backup-Lightweight-Container, und einen Optimierungs-Score für den Lightweight-Container und den entsprechenden Backup-Lightweight-Container identifizieren;

Trainieren, durch einen oder mehrere Prozessoren, eines Maschinenlernmodells unter Verwendung der identifizierten Trainingsdaten;

Identifizieren, durch einen oder mehrere Prozessoren, eines neuen Lightweight-Containers für Backup; und

Ermitteln, durch einen oder mehrere Prozessoren, eines optimalen Backup-Lightweight-Containers für den neuen Lightweight-Container unter Verwendung des trainierten Maschinenlernmodells.

2. Computerimplementiertes Verfahren nach Anspruch 1, wobei:
die Trainingsdaten ferner identifizieren: (i) unstrukturierte Container-Daten für den Lightweight-Container, (ii) Konfigurations-Daten für den Lightweight-Container, und (iii) Backup-Daten für den Backup-Lightweight-Container; und
der neue Lightweight-Container entsprechende unstrukturierte Containerdaten und Konfigurations-Daten beinhaltet.

3. Computerimplementiertes Verfahren nach Anspruch 2, wobei:
die unstrukturierten Container-Daten Textdaten für den Lightweight-Container beinhalten, wobei die Textdaten mindestens einen Containernamen und Log-Daten beinhalten;
die Konfigurations-Daten Hardware- und Software-Anforderungen für den Lightweight-Container beinhalten; und
die Backup-Daten Hardware- und Software-Merkmale für den Backup-Lightweight-Container beinhalten.

4. Computerimplementiertes Verfahren nach Anspruch 1, wobei der ermittelte optimale Backup-Lightweight-Container auf einer Mehrzahl von Knoten in einer Container-Umgebung läuft, wobei die Mehrzahl von Knoten einen Master-Knoten und einen oder mehrere Worker-Knoten enthält.

5. Computerimplementiertes Verfahren nach Anspruch 2, wobei das Verfahren ferner umfasst:
Identifizieren, durch einen oder mehrere Prozessoren, eines oder mehrerer Muster aus den unstrukturierten Container-Daten für den Lightweight-Container; und
Einbeziehen, durch einen oder mehrere Prozessoren, des identifizierten einen oder mehrerer Muster in das Trainieren des Maschinenlernmodells,
wobei der ermittelte optimale Backup-Lightweight-Container für den neuen Lightweight-Container zumindest teilweise auf Grundlage der jeweiligen Muster aus den unstrukturierten Container-Daten des neuen Lightweight-Containers ermittelt wird.

6. Computerimplementiertes Verfahren nach Anspruch 5, wobei die Identifizierung des einen oder der mehreren Muster einen K-Mean-Algorithmus verwendet.

7. Computerimplementiertes Verfahren nach Anspruch 6, ferner umfassend:
Ermitteln, durch einen oder mehrere Prozessoren, des Optimierungs-Scores für den Lightweight-Container und den entsprechenden Backup-Lightweight-Container, wobei der Optimierungs-Score zumindest teilweise basiert auf: (i) einem Ergebnis des K-Mean-Algorithmus, (ii) einer Latenzbeschränkung und (iii) einer erforderlichen Backup-Zeitspanne.

8. Computerprogrammprodukt, umfassend:
ein oder mehrere computerlesbare Speichermedien und Programmanweisungen, die auf dem einen oder den mehreren computerlesbaren Speichermedien gespeichert sind, wobei die gespeicherten Programmanweisungen umfassen:
Programmanweisungen zum Identifizieren eines Satzes von Trainingsdaten, wobei die Trainingsdaten einen Lightweight-Container, einen entsprechenden Backup-Lightweight-Container, und einen Optimierungs-Score für den Lightweight-Container und den entsprechenden Backup-Lightweight-Container identifizieren;
Programmanweisungen zum Trainieren eines Maschinenlernmodells unter Verwendung der identifizierten Trainingsdaten;
Programmanweisungen zum Identifizieren eines neuen Lightweight-Containers für Backup; und
Programmanweisungen zum Ermitteln eines optimalen Backup-Lightweight-Containers für den neuen Lightweight-Container unter Verwendung des trainierten Maschinenlernmodells.

9. Computerprogrammprodukt nach Anspruch 8, wobei:
die Trainingsdaten ferner identifizieren: (i) unstrukturierte Container-Daten für den Lightweight-Container, (ii) Konfigurations-Daten für den Lightweight-Container, und (iii) Backup-Daten für den Backup-Lightweight-Container; und
der neue Lightweight-Container entsprechende unstrukturierte Container-Daten und Konfigurations-Daten beinhaltet.

10. Computerprogrammprodukt nach Anspruch 9, wobei:
die unstrukturierten Container-Daten Textdaten für den Lightweight-Container beinhalten, wobei die Textdaten mindestens einen Containernamen und Log-Daten beinhalten;
die Konfigurations-Daten Hardware- und Software-Anforderungen für den Lightweight-Container beinhalten;

und
die Backup-Daten Hardware- und Software-Merkmale für den Backup-Lightweight-Container beinhalten.

11. Computerprogrammprodukt nach Anspruch 8, wobei der ermittelte optimale Backup-Lightweight-Container auf einer Mehrzahl von Knoten in einer Container-Umgebung läuft, wobei die Mehrzahl von Knoten einen Master-Knoten und einen oder mehrere Worker-Knoten enthält.

12. Computerprogrammprodukt nach Anspruch 9, wobei die gespeicherten Programmanweisungen ferner umfassen:
Programmanweisungen zum Identifizieren eines oder mehrerer Muster aus den unstrukturierten Containerdaten für den Lightweight-Container; und
Programmanweisungen zum Einbeziehen des identifizierten einen oder mehrerer Muster zum Trainieren des Maschinenlernmodells,
wobei der ermittelte optimale Backup-Lightweight-Container für den neuen Lightweight-Container zumindest teilweise auf Grundlage der jeweiligen Muster aus den unstrukturierten Container-Daten des neuen Lightweight-Containers ermittelt wird.

13. Computerprogrammprodukt nach Anspruch 12, wobei das Identifizieren des einen oder der mehreren Muster einen K-Mean-Algorithmus verwendet.

14. Computerprogrammprodukt nach Anspruch 13, wobei die gespeicherten Programmanweisungen ferner umfassen:
Programmanweisungen zum Ermitteln des Optimierungs-Scores für den Lightweight-Container und den entsprechenden Backup-Lightweight-Container, wobei der Optimierungs-Score zumindest teilweise basiert auf (i) einem Ergebnis des K-Mean-Algorithmus, (ii) einer Latenzbeschränkung und (iii) einer erforderlichen Backup-Zeitspanne.

15. Datenverarbeitungssystem, wobei das Computersystem umfasst:
einen oder mehrere Computer-Prozessoren:
ein oder mehrere computerlesbare Speichermedien; und
Programmanweisungen, die auf dem computerlesbaren Speichermedium zur Ausführung durch mindestens einen des einen oder der mehreren Computer-Prozessoren gespeichert sind, wobei die gespeicherten Programmanweisungen umfassen:
Programmanweisungen zum Identifizieren eines Satzes von Trainingsdaten, wobei die Trainingsdaten einen Lightweight-Container, einen entsprechenden Backup-Lightweight-Container, und einen Optimierungs-Score für den Lightweight-Container und den entsprechenden Backup-Lightweight-Container identifizieren;
Programmanweisungen zum Trainieren eines Maschinenlernmodells unter Verwendung der identifizierten Trainingsdaten;
Programmanweisungen zum Identifizieren eines neuen Lightweight-Containers für Backup; und
Programmanweisungen zum Ermitteln eines optimalen Backup-Lightweight-Containers für den neuen Lightweight-Container unter Verwendung des trainierten Maschinenlernmodells.

16. Computersystem nach Anspruch 15, wobei:
die Trainingsdaten ferner identifizierten: (i) unstrukturierte Container-Daten für den Lightweight-Container, (ii) Konfigurations-Daten für den Lightweight-Container, und (iii) Backup-Daten für den Backup-Lightweight-Container; und
der neue Lightweight-Container entsprechende unstrukturierte Container-Daten und Konfigurations-Daten beinhaltet.

17. Computersystem nach Anspruch 16, wobei:
die unstrukturierten Containerdaten Textdaten für den Lightweight-Container beinhalten, wobei die Textdaten mindestens einen Containernamen und Log-Daten beinhalten;
die Konfigurations-Daten Hardware- und Software-Anforderungen für den Lightweight-Container beinhalten; und
die Backup-Daten Hardware- und Software-Merkmale für den Backup-Lightweight-Container beinhalten.

18. Computersystem nach Anspruch 16, wobei die gespeicherten Programmanweisungen ferner umfassen:
Programmanweisungen zum Identifizieren eines oder mehrerer Muster aus den unstrukturierten Containerdaten für den Lightweight-Container; und

Programmanweisungen zum Einbeziehen des identifizierten einen oder mehrerer Muster in das Trainieren des Maschinenlernmodells,
wobei der ermittelte optimale Backup-Lightweight-Container für den neuen Lightweight-Container zumindest teilweise auf Grundlage der jeweiligen Muster aus den unstrukturierten Container-Daten des neuen Lightweight-Containers ermittelt wird.

19. Computersystem nach Anspruch 18, wobei das Identifizieren des einen oder der mehreren Muster einen K-Mean-Algorithmus verwendet.

20. Computersystem nach Anspruch 19, wobei die gespeicherten Programmanweisungen ferner umfassen:
Ermitteln, durch einen oder mehrere Prozessoren, des Optimierungs-Scores für den Lightweight-Container und den entsprechenden Backup-Lightweight-Container, wobei der Optimierungs-Score zumindest teilweise basiert auf: (i) einem Ergebnis des K-Mean-Algorithmus, (ii) einer Latenzbeschränkung und (iii) einer erforderlichen Backup-Zeitspanne.

Es folgen 7 Seiten Zeichnungen

Anhängende Zeichnungen

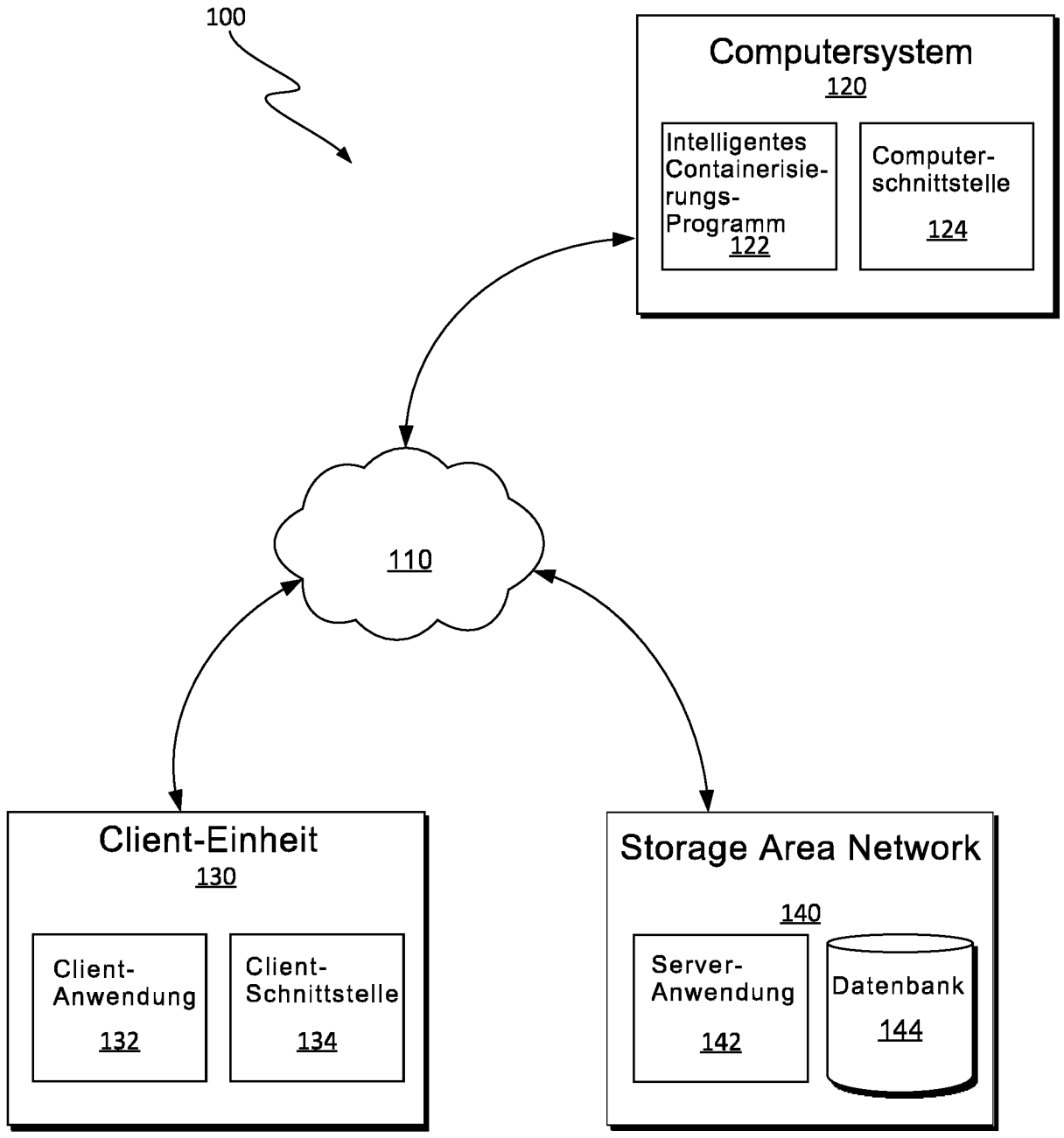


FIG. 1

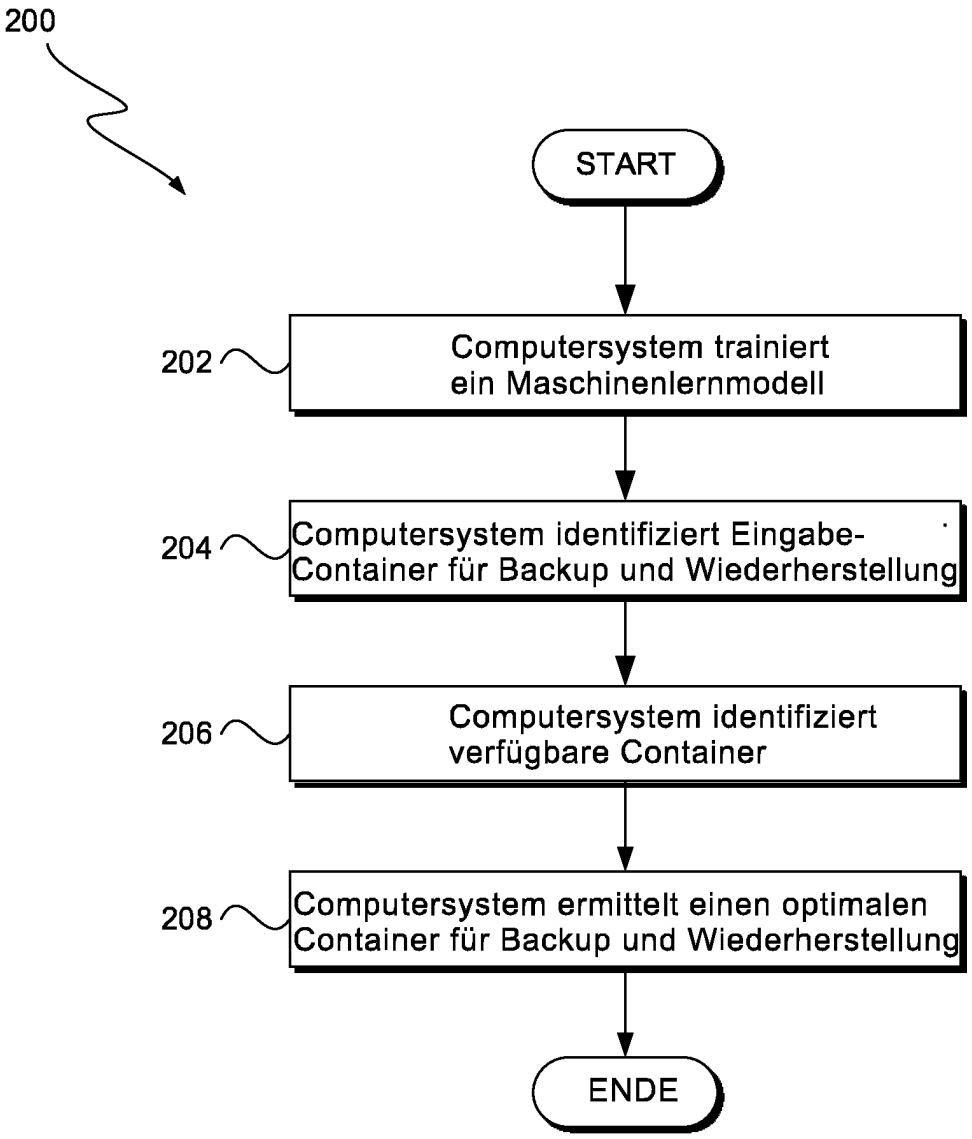

200
START
202
Computersystem trainiert ein Maschinenlernmodell
204
Computersystem identifiziert Eingabe-Container für Backup und Wiederherstellung
206
Computersystem identifiziert verfügbare Container
208
Computersystem ermittelt einen optimalen Container für Backup und Wiederherstellung
ENDE

FIG. 2

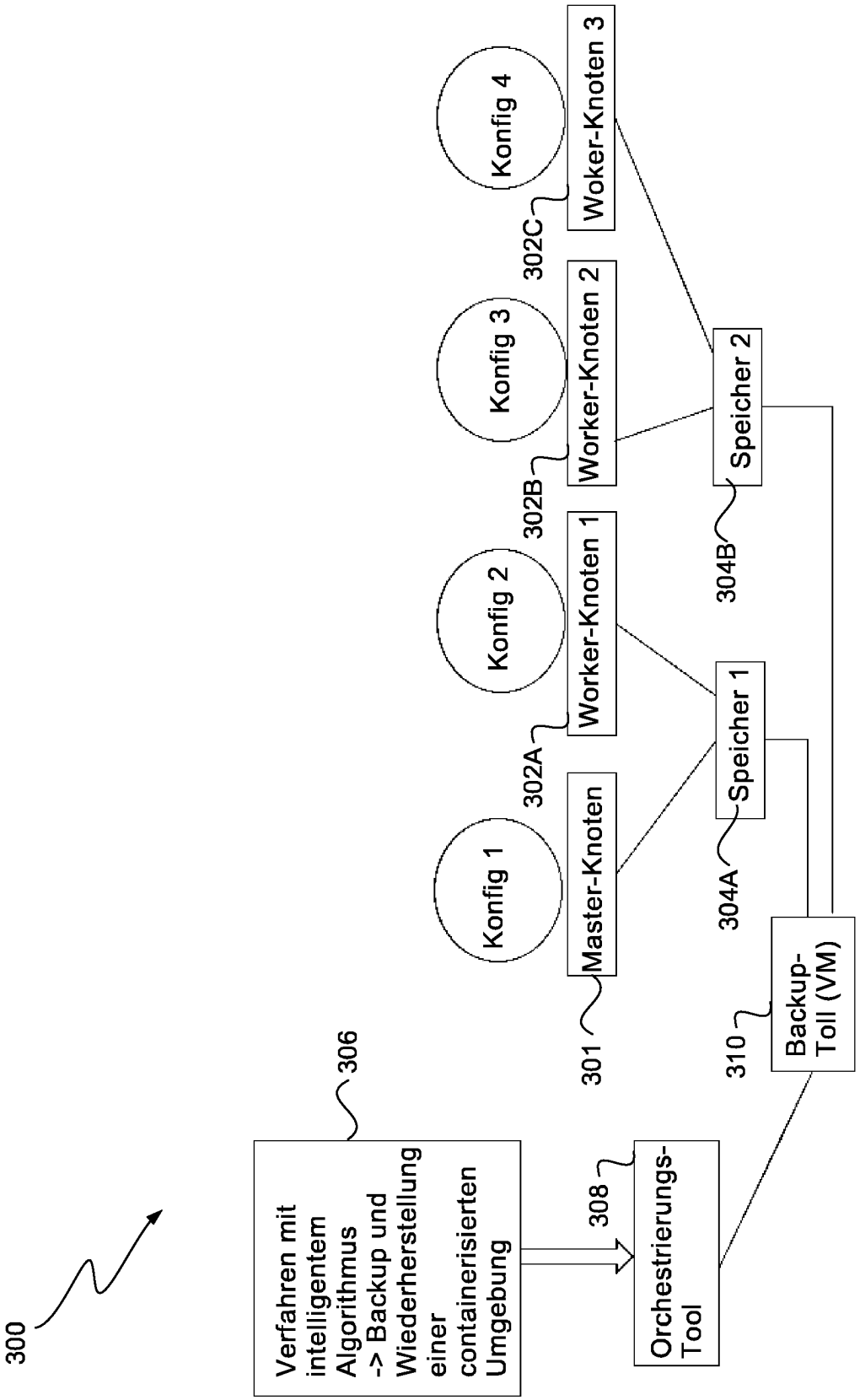

300
Verfahren mit intelligentem Algorithmus -> Backup und Wiederherstellung einer containerisierten Umgebung
306
308
Orchestrierungs-Tool
Backup-Toll (VM)
310
Master-Knoten
301
Worker-Knoten 1
302A
Worker-Knoten 2
302B
Woker-Knoten 3
302C
Konfig 1
Konfig 2
Konfig 3
Konfig 4
Speicher 1
304A
Speicher 2
304B

FIG. 3

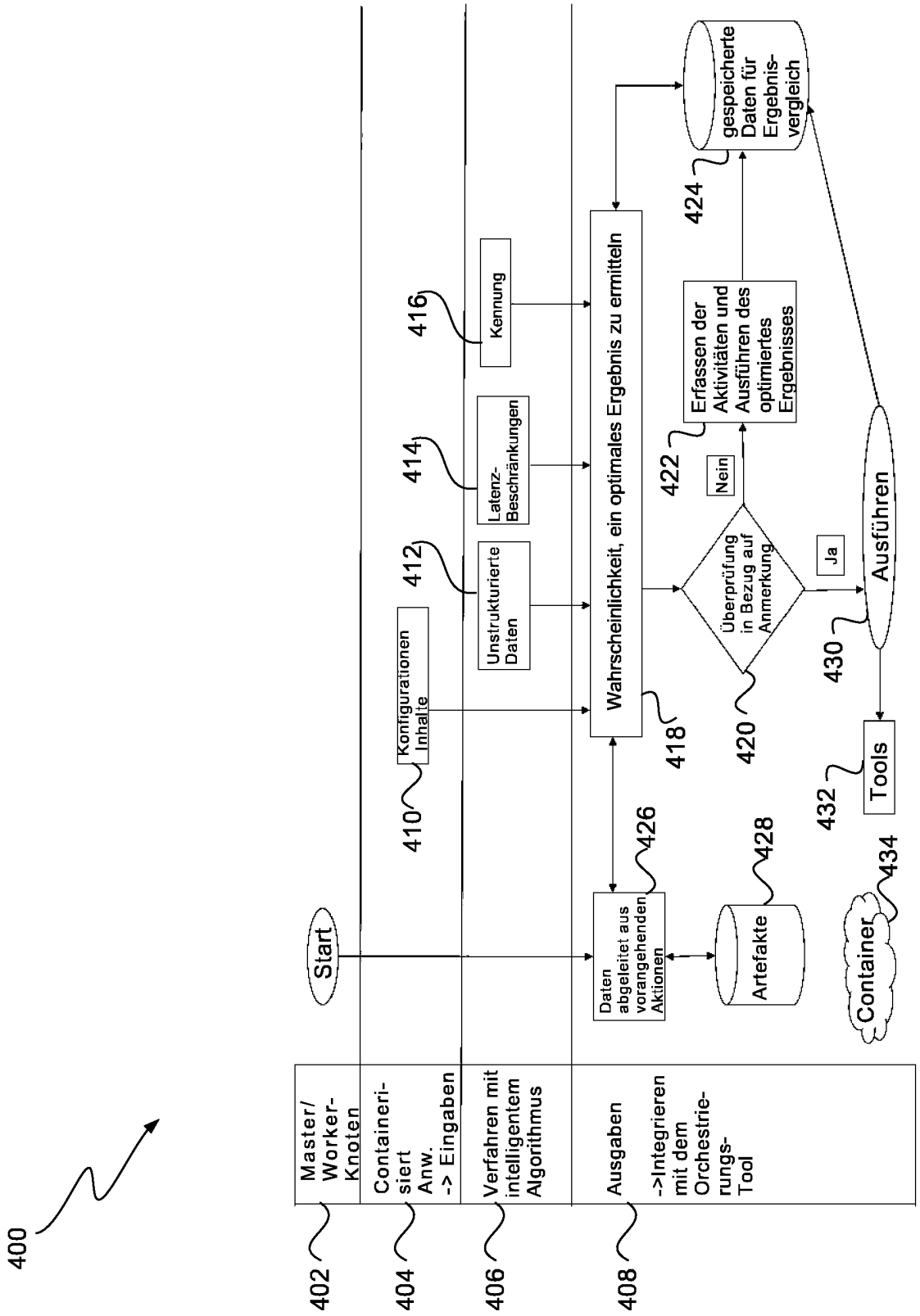

400
Master/ Worker-Knoten
Containeri-siert Anw. -> Eingaben
Verfahren mit intelligentem Algorithmus
Ausgaben ->Integrieren mit dem Orchestrie-rungs-Tool
402
404
406
408
Start
Konfigurationen Inhalte
410
Unstrukturierte Daten
412
Latenz-Beschränkungen
414
Kennung
416
Daten abgeleitet aus vorangehenden Aktionen
426
Artefakte
428
Container
434
Wahrscheinlichkeit, ein optimales Ergebnis zu ermitteln
418
Überprüfung in Bezug auf Anmerkung
420
Nein
Ja
422
Erfassen der Aktivitäten und Ausführen des optimiertes Ergebnisses
gespeicherte Daten für Ergebnis-vergleich
424
Ausführen
430
Tools
432

FIG. 4

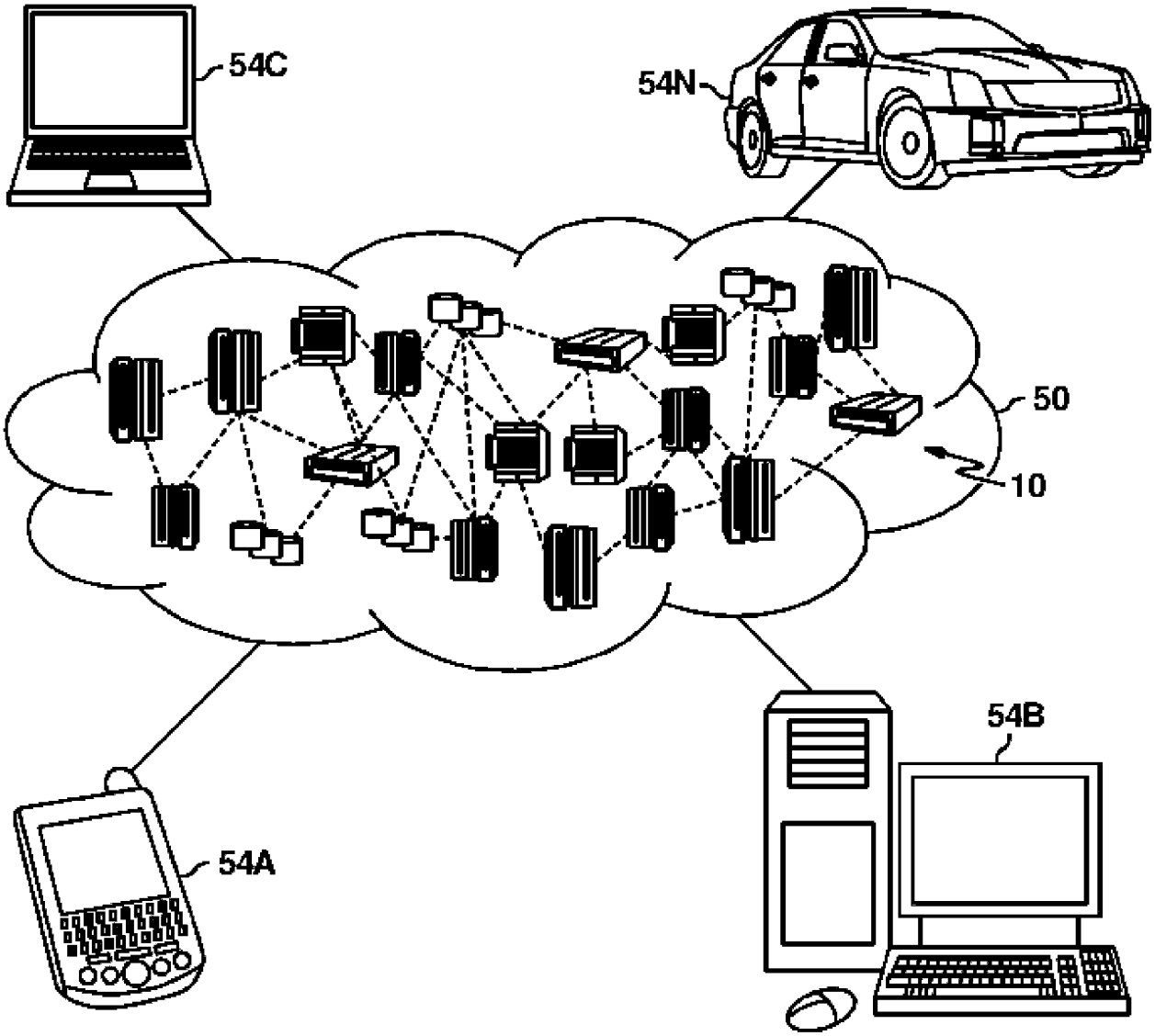


FIG. 5

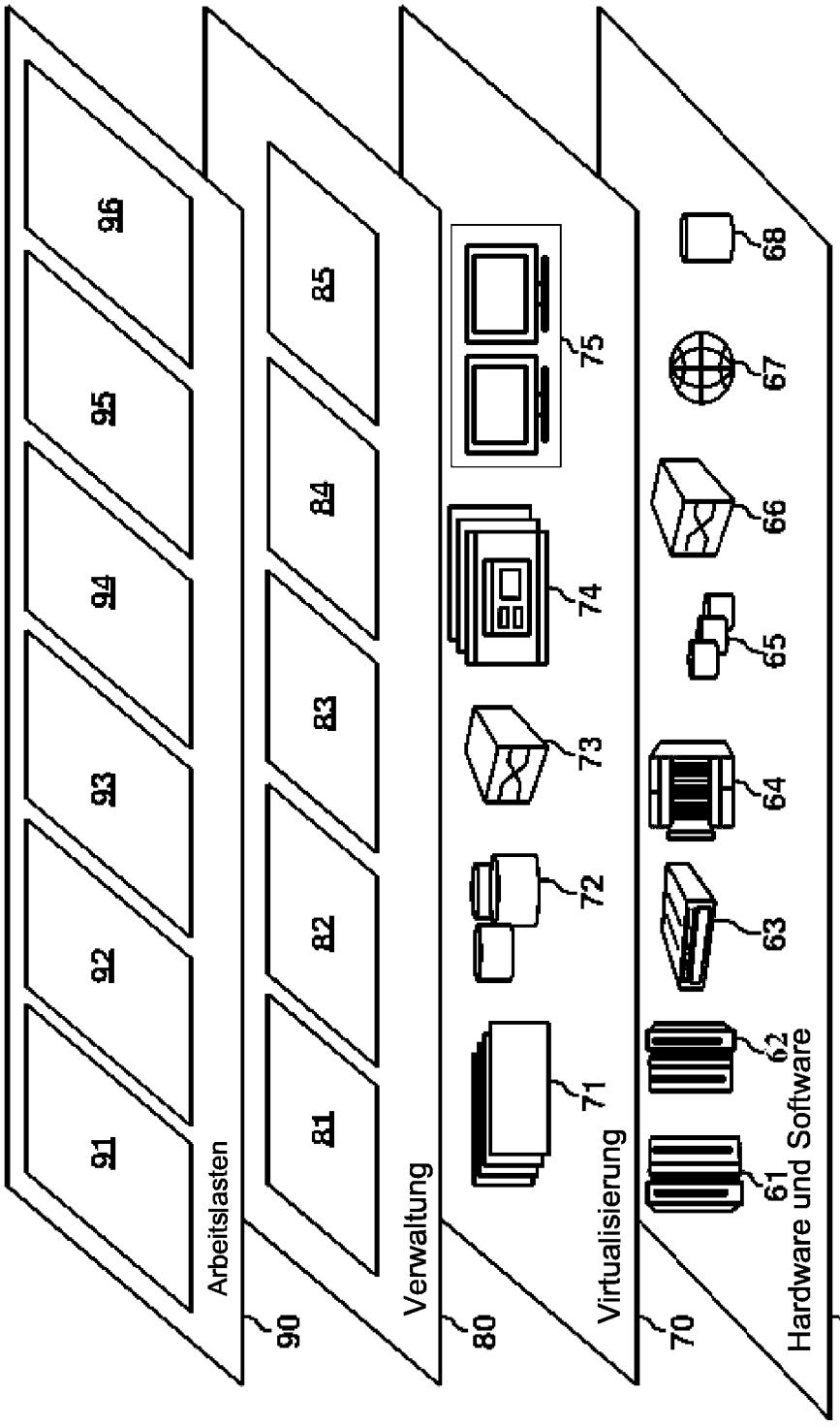

Arbeitslasten
90
91
92
93
94
95
96
Verwaltung
80
81
82
83
84
85
Virtualisierung
70
71
72
73
74
75
Hardware und Software
60
61
62
63
64
65
66
67
68

FIG. 6

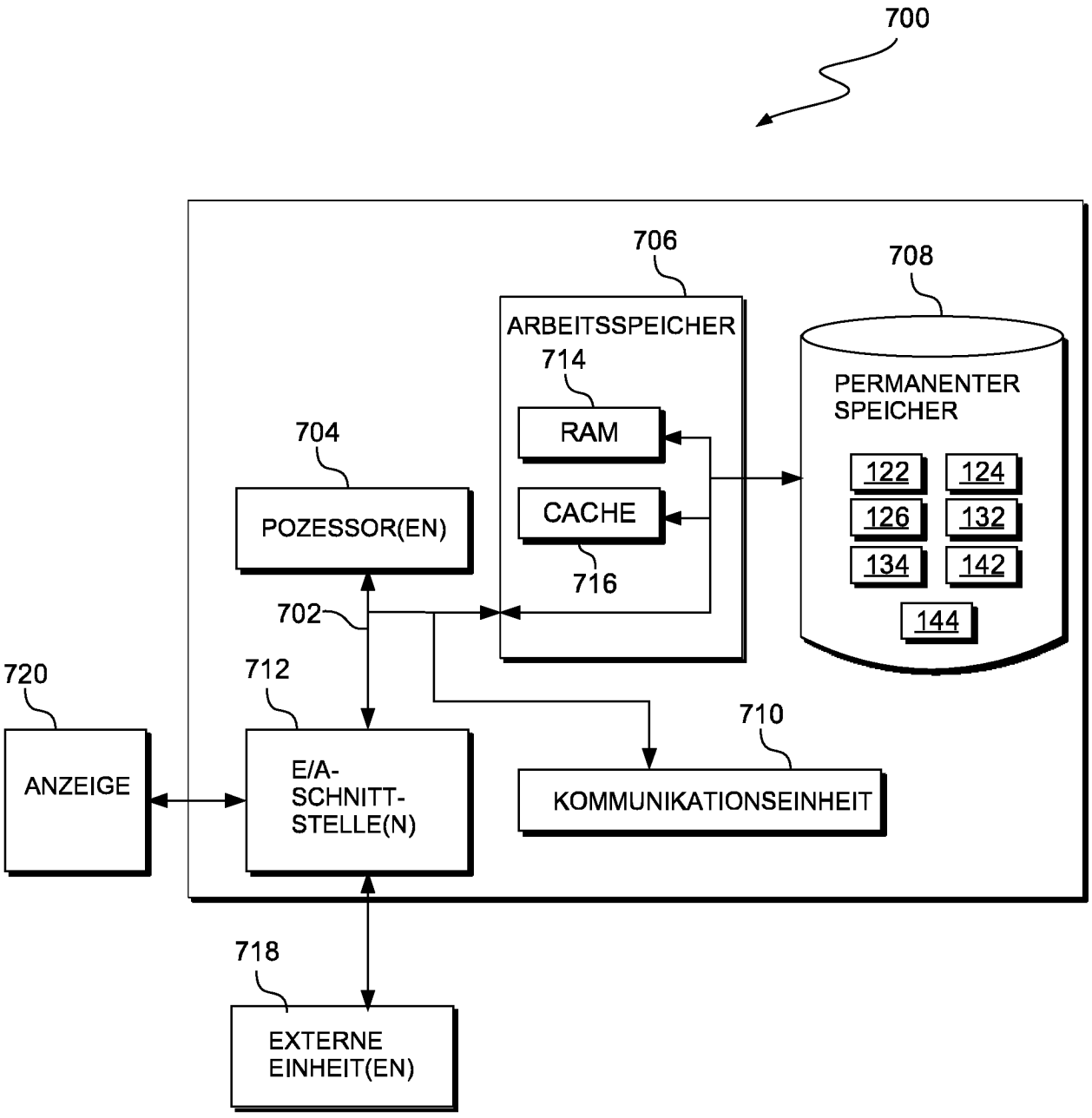


FIG. 7

 Das Dokument wurde durch die Firma Luminess hergestellt.