



(12) **Patentschrift**

(21) Deutsches Aktenzeichen: **11 2021 003 525.8**
(86) PCT-Aktenzeichen: **PCT/CN2021/116202**
(87) PCT-Veröffentlichungs-Nr.: **WO 2022/048595**
(86) PCT-Anmeldetag: **02.09.2021**
(87) PCT-Veröffentlichungstag: **10.03.2022**
(43) Veröffentlichungstag der PCT Anmeldung
in deutscher Übersetzung: **06.07.2023**
(45) Veröffentlichungstag
der Patenterteilung: **02.10.2024**

(51) Int Cl.: **G06F 9/50 (2006.01)**
G06F 11/34 (2006.01)
G10L 15/26 (2006.01)

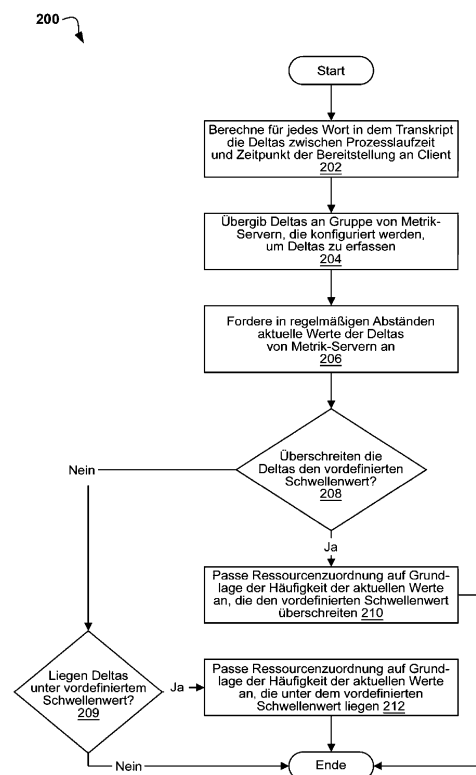
Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz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

(30) Unionspriorität: 17/010,866 03.09.2020 US	(72) Erfinder: Bolanos, Daniel, Yorktown Heights, NY, US; Lee, Antonio Rogelio, Endicott, NY, US
(73) Patentinhaber: International Business Machines Corporation, Armonk, NY, US	(56) Ermittelte Stand der Technik: US 9 098 338 B2 US 2007 / 0 143 116 A1 US 2010 / 0 312 556 A1 US 2016 / 0 292 000 A1
(74) Vertreter: Richardt Patentanwälte PartG mbB, 65185 Wiesbaden, DE	

(54) Bezeichnung: **AUTOMATISCHE SPRACHE-ZU-TEXT-SKALIERUNG FÜR LIVE-ANWENDUNGSFÄLLE**

(57) Hauptanspruch: Computergestütztes Verfahren zum Zuordnen von Rechenressourcen, wobei das Verfahren aufweist:

Berechnen eines Deltas zwischen einer Prozesslaufzeit und einem Zeitpunkt, zu dem das Wort einem Client bereitgestellt wird, für jedes Wort in einem Transkript;
Übergeben der Deltas an eine Gruppe von Metrik-Servern, wobei die Gruppe von Metrik-Servern konfiguriert wird, um die Deltas zu erfassen;
Anfordern von aktuellen Werten der Deltas von der Gruppe von Metrik-Servern;
Ermitteln, ob die aktuellen Werte der Deltas einen vordefinierten Schwellenwert für eine maximale Latenz, Max-Latenz-Schwellenwert, überschreiten; und
als Reaktion auf ein Ermitteln, dass die aktuellen Werte der Deltas den vordefinierten Max-Latenz-Schwellenwert überschreiten, Anpassen der zugeordneten Rechenressourcen auf Grundlage einer Häufigkeit der aktuellen Werte der Deltas, die den vordefinierten Max-Latenz-Schwellenwert überschreiten.



Beschreibung**HINTERGRUND**

[0001] Die vorliegende Erfindung bezieht sich im Allgemeinen auf das Gebiet der Datenverarbeitung und im Besonderen auf ein automatisches Sprache-zu-Text-Skalierungssystem.

[0002] Sprache-zu-Text bezieht sich auf das Gebiet eines Transkribierens von Wörtern in eine Textform auf Grundlage von gesprochener Sprache. Reale Anwendungsfälle können Untertitel, automatisierte Agenten und Diktate enthalten, ohne jedoch darauf beschränkt zu sein. Bei dieser Art von Fällen ist eine natürliche und reibungslose Interaktion zwischen einem Benutzer und einem Sprache-zu-Text-Dienst auf Cloud-Grundlage erwünscht. Eine Live-Interaktion zwischen dem Benutzer und dem Sprache-zu-Text-Dienst auf Cloud-Grundlage erfordert eine schnelle und effiziente Bereitstellung von Erkennungstranskripten durch eine Spracherkennungs-Engine. Es ist wünschenswert, dass Backend-Server, die in einem entfernt angeordneten Rechenzentrum ausgeführt werden, genügend Spielraum haben, d.h. verfügbare Rechenressourcen, um dynamischen CPU-Anforderungen Central Processing Unit, Zentraleinheit) der Spracherkennungs-Engine gerecht zu werden. Es kann daher zwingend notwendig sein, über ein System zu verfügen, um Metriken, die mit einer Benutzererfahrung und den Rechenressourcen korrelieren, die zu einem beliebigen Zeitpunkt benötigt werden, dynamisch zu erzeugen und so die Rechenressourcen zuzuordnen und eine Latenz aufrechtzuerhalten.

[0003] US 9 098 338 B2 beschreibt ein Verfahren, beinhaltend: Verarbeiten eines Workflows mit mehreren Aufgaben durch Auswahl, Abrufen von Aufgabeninformationen und Ausführung basierend auf den übersetzten Informationen. US 2007 / 0 143 116 A1 beschreibt einen maschinenlesbaren Speicher, der einen Satz von Anweisungen zum Lastenausgleich enthalten kann. US 2010 / 0 312 556 A1 beschreibt Systeme, computerimplementierte Verfahren und greifbare computerlesbare Speichermedien zur Personalisierung der Sprechererkennung. US 2016 / 0 292 000 A1 beschreibt ein Verfahren, System und Computerprogrammprodukt zur Auswahl eines Ziel-Hypervisors zum Ausführen einer migrierten virtuellen Maschine.

KURZDARSTELLUNG

[0004] Die Erfindung wird durch die Merkmale der unabhängigen Ansprüche beschrieben. Ausführungsformen sind in den abhängigen Ansprüchen angegeben.

[0005] Gemäß einer Ausführungsform werden ein Verfahren, Computersystem und Computerprogrammprodukt für eine automatische Sprache-zu-Text-Skalierung von Rechenressourcen bereitgestellt. Die Ausführungsform kann ein Berechnen eines Deltas für jedes Wort in einem Transkript enthalten. Das berechnete Delta kann für jedes Wort in einem Transkript die Differenz zwischen einer Prozesslaufzeit und einem Zeitpunkt sein, zu dem das Wort einem Client bereitgestellt wird. Die Ausführungsform kann außerdem ein Übergeben der Deltas an eine Gruppe von Metrik-Servern enthalten, wobei die Metrik-Server konfiguriert werden, um die Deltas zu erfassen. Die Ausführungsform kann des Weiteren ein Anfordern von aktuellen Werten der Deltas von der Gruppe von Metrik-Servern enthalten. Die Ausführungsform kann außerdem ein Ermitteln enthalten, ob die aktuellen Werte der Deltas einen vordefinierten Schwellenwert für eine maximale Latenz (Max-Latenz-Schwellenwert) überschreiten. Die Ausführungsform kann des Weiteren ein Anpassen der zugeordneten Rechenressourcen auf Grundlage einer Häufigkeit der aktuellen Werte der Deltas enthalten, die den vordefinierten Max-Latenz-Schwellenwert überschreiten. Die Ausführungsform kann außerdem ein Erzeugen eines Histogramms aus den aktuellen Werten der Deltas enthalten. Die Ausführungsform kann als Reaktion auf ein Erzeugen des Histogramms des Weiteren ein Durchführen einer Aufwärtsskalierung der zugeordneten Rechenressourcen auf Grundlage eines Prozentsatzes von Datenpunkten enthalten, die über dem vordefinierten Max-Latenz-Schwellenwert liegen.

KURZBESCHREIBUNG DER VERSCHIEDENEN ANSICHTEN DER ZEICHNUNGEN

[0006] Diese und andere Ziele, Merkmale und Vorzüge der vorliegenden Erfindung werden aus der folgenden ausführlichen Beschreibung von veranschaulichenden Ausführungsformen hiervon offensichtlich, die in Verbindung mit den beigefügten Zeichnungen zu lesen ist. Die verschiedenen Merkmale der Zeichnungen sind nicht maßstäblich, da die Darstellungen der Klarheit dienen, indem sie einem Fachmann dabei helfen, die Erfindung in Verbindung mit der ausführlichen Beschreibung zu verstehen. In den Zeichnungen:

veranschaulicht **Fig. 1** eine beispielhafte vernetzte Computerumgebung gemäß mindestens einer Ausführungsform;

veranschaulicht **Fig. 2** einen Betriebsablaufplan zum Zuordnen von Rechenressourcen in einem Prozess für eine automatische Sprache-zu-Text-Skalierung gemäß mindestens einer Ausführungsform;

ist **Fig. 3** ein funktionales Blockschaubild eines in Software realisierten Qualitätsbewertungs-

prozesses gemäß mindestens einer Ausführungsform;

stellt **Fig. 4** eine Cloud-Computing-Umgebung gemäß einer Ausführungsform der vorliegenden Erfindung dar;

stellt **Fig. 5** Abstraktionsmodellschichten gemäß einer Ausführungsform der vorliegenden Erfindung dar.

AUSFÜHRLICHE BESCHREIBUNG

[0007] Hierin werden detaillierte Ausführungsformen der beanspruchten Strukturen und Verfahren offenbart; dabei sollte allerdings klar sein, dass die offenbarten Ausführungsformen die beanspruchten Strukturen und Verfahren, die auf verschiedene Arten ausgeführt werden können, lediglich veranschaulichen. Diese Erfindung kann jedoch auf viele unterschiedliche Arten ausgeführt werden und sollte nicht als auf die hier genannten, beispielhaften Ausführungsformen beschränkt verstanden werden. In der Beschreibung können Einzelheiten von hinlänglich bekannten Merkmalen und Techniken weggelassen werden, um zu verhindern, dass die dargelegten Ausführungsformen unnötigerweise unklar gemacht werden.

[0008] Ausführungsformen der vorliegenden Erfindung beziehen sich auf das Gebiet der Datenverarbeitung und im Besonderen auf ein automatisches Sprache-zu-Text-Skalierungssystem. Die im Folgenden beschriebenen beispielhaften Ausführungsformen stellen ein System, Verfahren und Programmprodukt bereit, um unter anderem eine Sprache-zu-Text-Verarbeitung und eine Verarbeitung natürlicher Sprache (Natural Language Processing, NLP) zum dynamischen Erzeugen von Metriken zu verwenden, die mit der Benutzererfahrung und den Rechenressourcen korrelieren, die zu einem beliebigen Zeitpunkt benötigt werden, und die Rechenressourcen, die notwendig sind, um eine Latenz innerhalb eines akzeptablen Bereichs zu halten, entsprechend zuzuordnen. Dabei sollte klar sein, dass „Rechenressourcen“ und „Backend-Server“ hierin austauschbar verwendet werden. Die vorliegende Ausführungsform hat somit die Fähigkeit, das technische Gebiet einer automatischen Sprache-zu-Text-Skalierung zu verbessern, indem eine natürliche und reibungslose Interaktion zwischen dem Benutzer und einem Sprache-zu-Text-Dienst auf Cloud-Grundlage ermöglicht wird.

[0009] Wie weiter oben beschrieben, bezieht sich Sprache-zu-Text auf das Gebiet eines Transkribierens von Wörtern in eine Textform auf Grundlage von gesprochener Sprache. Echtzeitgarantien und insbesondere eine durch den Benutzer wahrgenommene Latenz sind von entscheidender Bedeutung für die Benutzererfahrung in allen Sprache-zu-Text-

Anwendungsfällen, die eine Live-Interaktion zwischen dem Benutzer und dem Sprache-zu-Text-Dienst auf Cloud-Grundlage einbeziehen. Reale Anwendungsfälle können Untertitel, automatisierte Agenten und Diktate enthalten, ohne jedoch darauf beschränkt zu sein. Bei dieser Art von Fällen müssen Erkennungstranskripte mit geringer Verzögerung durch eine Spracherkennungs-Engine bereitgestellt werden. Allerdings tragen zwischen einer Audioquelle und der Spracherkennungs-Engine, die in einem entfernt angeordneten Rechenzentrum ausgeführt wird, mehrere Faktoren zu einer Latenzverschlechterung bei, z.B. Backend-Server, die nicht genügend Spielraum in Bezug auf verfügbare Rechenressourcen haben, um den dynamischen CPU-Anforderungen der Spracherkennungs-Engine gerecht zu werden. Dieses Problem wird üblicherweise gelöst, indem ein Übermaß an Rechenressourcen zugeordnet wird, um ein Worst-Case-Szenario eines schwierig zu verarbeitenden Datenstroms abzudecken. Die Menge an Rechenressourcen, üblicherweise in CPU-Form, die die Spracherkennungs-Engine benötigt, um einen Strom von Transkripten in Echtzeit mit geringer Latenz zu erzeugen, variiert stark zwischen den verschiedenen Audioströmen, die durch die Backend-Server verarbeitet werden. Ohne darauf beschränkt zu sein, können Faktoren, die mit der Menge der benötigten Rechenressourcen in Zusammenhang stehen, Sprechermerkmale, Stil, Übersprechen, Stillebereiche, Hintergrundgeräusche, Datenverkehrsmuster und kollokierte Arbeitslasten enthalten, d.h. andere Anwendungen oder andere Instanzen der Spracherkennungs-Engine, die auf derselben Host-Maschine ausgeführt werden und um Rechenressourcen konkurrieren. Es kann daher unter anderem vorteilhaft sein, proaktiv Maßnahmen zu ergreifen, um diese Faktoren zu berücksichtigen, indem eine Anzahl der Backend-Server automatisch aufwärts oder abwärts skaliert wird, sodass die verfügbaren Rechenressourcen auf einen Datenverkehr abgestimmt sind, der von den Benutzern eingeht. Somit können Rechenressourcen auf Grundlage von Notwendigkeit und Latenz zugeordnet, während Kosten innerhalb des zulässigen Bereichs gehalten werden.

[0010] Gemäß mindestens einer Ausführungsform kann für jedes Wort in dem Erkennungstranskript ein Delta berechnet werden. Das berechnete Delta kann eine Zeitdifferenz zwischen einer Prozesslaufzeit, wenn Audiosignale für das Wort an die Spracherkennungs-Engine gesendet werden, und dem Zeitpunkt sein, zu dem das Wort einem Client, d.h. einer Anwendungsschicht, bereitgestellt wird. Die Deltas können an eine Gruppe von Metrik-Servern übergeben werden, die konfiguriert werden, um die Deltas zu erfassen. Eine Gruppe von automatischen Horizontalskalierern kann in regelmäßigen Abständen aktuelle Werte der Deltas von den Metrik-Servern anfordern. Wenn die aktuellen Werte der Deltas

einen vordefinierten Schwellenwert für eine maximale Latenz überschreiten, kann bei mindestens einer Ausführungsform die Gruppe von automatischen Horizontalskalierern eine Aufwärtsskalierungsoperation auslösen und der Spracherkennungs-Engine, die in dem entfernt angeordneten Rechenzentrum ausgeführt wird, weitere Rechenressourcen hinzufügen. Die Rechenressourcen können inkrementell so lange hinzugefügt werden, bis die aktuellen Werte der Deltas innerhalb des akzeptablen Bereichs für eine Latenz liegen. Wenn die aktuellen Werte der Deltas unter einem vordefinierten Schwellenwert für eine minimale Latenz (Min-Latenz-Schwellenwert) liegen, kann bei mindestens einer anderen Ausführungsform die Gruppe von automatischen Horizontalskalierern eine Abwärtsskalierungsoperation auslösen und die Rechenressourcen reduzieren, die der Spracherkennungs-Engine zugeordnet werden, die in dem entfernt angeordneten Rechenzentrum ausgeführt wird. Entsprechend können die Rechenressourcen inkrementell so lange reduziert werden, bis die aktuellen Werte der Deltas innerhalb des akzeptablen Bereichs für eine Latenz liegen.

[0011] Bei der vorliegenden Erfindung kann es sich um ein System, ein Verfahren und/oder ein Computerprogrammprodukt mit einem beliebigen möglichen Grad an technischer Integration handeln. Das Computerprogrammprodukt kann ein computerlesbares Speichermedium (oder computerlesbare Speichermedien) mit darauf gespeicherten computerlesbaren Programmanweisungen enthalten, um einen Prozessor dazu zu veranlassen, Aspekte der vorliegenden Erfindung auszuführen.

[0012] Bei dem computerlesbaren Speichermedium kann es sich um eine physische Einheit handeln, die Anweisungen zur Verwendung durch eine Anweisungsausführungseinheit behalten und speichern kann. Bei dem computerlesbaren Speichermedium kann es sich zum Beispiel um eine elektronische Speichereinheit, eine magnetische Speichereinheit, eine optische Speichereinheit, eine elektromagnetische Speichereinheit, eine Halbleiterspeichereinheit oder jede geeignete Kombination daraus handeln, ohne auf diese beschränkt zu sein. Zu einer nicht erschöpfenden Liste spezifischerer Beispiele des computerlesbaren Speichermediums gehören die Folgenden: eine tragbare Computerdiskette, eine Festplatte, ein Direktzugriffsspeicher (Random Access Memory, RAM), ein Nur-Lese-Speicher (ROM) ein löschbarer programmierbarer Nur-Lese-Speicher (EEPROM bzw. Flash-Speicher), ein statischer Direktzugriffsspeicher (SRAM), ein tragbarer CD-ROM, eine DVD (Digital Versatile Disc), ein Speicher-Stick, eine Diskette, eine mechanisch kodierte Einheit wie zum Beispiel Lochkarten oder erhabene Strukturen in einer Rille, auf denen Anweisungen gespeichert werden, und jede geeignete Kombina-

tion daraus. Ein computerlesbares Speichermedium soll in der Verwendung hierin nicht als flüchtige Signale an sich aufgefasst werden, wie zum Beispiel Funkwellen oder andere sich frei ausbreitende elektromagnetische Wellen, elektromagnetische Wellen, die sich durch einen Wellenleiter oder ein anderes Übertragungsmedium ausbreiten (z.B. durch ein Lichtwellenleiterkabel geleitete Lichtimpulse) oder durch einen Draht übertragene elektrische Signale.

[0013] Hierin beschriebene computerlesbare Programmanweisungen können von einem computerlesbaren Speichermedium auf jeweilige Datenverarbeitungs-/ Verarbeitungseinheiten oder über ein Netzwerk wie zum Beispiel das Internet, ein lokales Netzwerk (Local Area Network, LAN), ein Weitverkehrsnetzwerk (Wide Area Network, WAN) 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 computerlesbare Programmanweisungen aus dem Netzwerk und leitet die computerlesbaren Programmanweisungen zur Speicherung in einem computerlesbaren Speichermedium innerhalb der entsprechenden Datenverarbeitungs-/ Verarbeitungseinheit weiter.

[0014] Bei computerlesbaren 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, zustandseinstellende Daten, Konfigurationsdaten für eine integrierte Schaltung oder sowohl um Quellcode als auch um Objektcode handeln, die in einer beliebigen Kombination aus einer oder mehreren Programmiersprachen geschrieben werden, darunter objektorientierte Programmiersprachen wie Smalltalk, C++ o.ä. sowie herkömmliche prozedurale Programmiersprachen wie die Programmiersprache „C“ oder ähnliche Programmiersprachen. Die computerlesbaren 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 letzterem Fall kann der entfernt angeordnete Computer mit dem Computer des Benutzers durch eine beliebige Art Netzwerk verbunden werden, zum Beispiel ein lokales Netzwerk (LAN) oder ein Weitverkehrsnetzwerk (WAN), oder die Verbindung kann mit einem exter-

nen Computer hergestellt werden (zum Beispiel über das Internet unter Verwendung eines Internet-Diensteanbieters). Bei manchen Ausführungsformen können elektronische Schaltungen, darunter zum Beispiel programmierbare Logikschaltungen, im Feld programmierbare Gatter-Arrays (Field-Programmable Gate Arrays, FPGA) oder programmierbare Logik-Arrays (Programmable Logic Arrays, PLA) die computerlesbaren Programmanweisungen ausführen, indem sie Zustandsinformationen der computerlesbaren Programmanweisungen nutzen, um die elektronischen Schaltungen zu personalisieren, um Aspekte der vorliegenden Erfindung durchzuführen.

[0015] Aspekte der vorliegenden Erfindung werden hierin unter Bezugnahme auf Ablaufpläne und/oder Blockschaubilder von Verfahren, Vorrichtungen (Systemen) und Computerprogrammprodukten gemäß Ausführungsformen der Erfindung beschrieben. Es wird darauf hingewiesen, dass jeder Block der Ablaufpläne und/oder der Blockschaubilder sowie Kombinationen von Blöcken in den Ablaufplänen und/oder den Blockschaubildern mittels computerlesbarer Programmanweisungen ausgeführt werden können.

[0016] Diese computerlesbaren Programmanweisungen können einem Prozessor eines Universalcomputers, eines Spezialcomputers oder einer anderen programmierbaren Datenverarbeitungsvorrichtung bereitgestellt werden, um eine Maschine zu erzeugen, sodass die über den Prozessor des Computers bzw. der anderen programmierbaren Datenverarbeitungsvorrichtung ausgeführten Anweisungen ein Mittel zur Umsetzung der in dem Block bzw. den Blöcken des Ablaufplans und/oder des Blockschaubilds angegebenen Funktionen/Schritte erzeugen. Diese computerlesbaren Programmanweisungen können auch auf einem computerlesbaren Speichermedium gespeichert werden, das einen Computer, eine programmierbare Datenverarbeitungsvorrichtung und/oder andere Einheiten so steuern kann, dass sie auf eine bestimmte Art funktionieren, sodass das computerlesbare Speichermedium, auf dem Anweisungen gespeichert werden, ein Herstellungsprodukt aufweist, darunter Anweisungen, die Aspekte der bzw. des in dem Block bzw. den Blöcken der Ablaufpläne und/oder Blockschaubilder angegebenen Funktion/Schritts umsetzen.

[0017] Die computerlesbaren Programmanweisungen können auch auf einen Computer, eine andere programmierbare Datenverarbeitungsvorrichtung oder eine andere Einheit geladen werden, um das Ausführen einer Reihe von Prozessschritten auf dem Computer bzw. der anderen programmierbaren Vorrichtung oder anderen Einheit zu verursachen, um einen computerrealisierten 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 Blockschaubilder festgelegten Funktionen/Schritte umsetzen.

[0018] Die Ablaufpläne und die 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 Blockschaubildern ein Modul, ein Segment oder einen Teil von Anweisungen darstellen, die eine oder mehrere ausführbare Anweisungen zum Realisieren der bestimmten logischen Funktion bzw. Funktionen aufweisen. In einigen alternativen Ausführungen können die in dem Block angegebenen Funktionen in einer anderen Reihenfolge als in den Figuren gezeigt stattfinden. Zwei nacheinander gezeigte Blöcke können zum Beispiel in Wirklichkeit im Wesentlichen gleichzeitig ausgeführt werden, oder die Blöcke können manchmal je nach entsprechender Funktionalität in umgekehrter Reihenfolge ausgeführt werden. Es ist ferner anzumerken, dass jeder Block der Blockschaubilder und/oder der Ablaufpläne sowie Kombinationen aus Blöcken in den Blockschaubildern und/oder den Ablaufplänen durch spezielle auf Hardware beruhende Systeme umgesetzt werden können, die die festgelegten Funktionen oder Schritte durchführen, oder Kombinationen aus Spezial-Hardware und Computeranweisungen ausführen.

[0019] Die im Folgenden beschriebenen beispielhaften Ausführungsformen stellen ein System, Verfahren und Programmprodukt bereit, um in Echtzeit zu ermitteln, ob die zugeordneten Rechenressourcen ausreichen, um eine Latenz innerhalb des akzeptablen Bereichs zu halten, oder ob die zugeordneten Ressourcen angepasst werden müssen.

[0020] Unter Bezugnahme auf **Fig. 1** wird eine beispielhafte vernetzte Computerumgebung 100 gemäß mindestens einer Ausführungsform dargestellt. Die vernetzte Computerumgebung 100 kann eine Client-Datenverarbeitungseinheit 102 und einen Server 112 enthalten, die über ein Datenübertragungsnetzwerk 114 miteinander verbunden werden. Gemäß mindestens einer Realisierung kann die vernetzte Computerumgebung 100 eine Mehrzahl von Client-Datenverarbeitungseinheiten 102 und Servern 112 enthalten, von denen aus Gründen der Kürze nur jeweils eine bzw. einer gezeigt wird.

[0021] Das Datenübertragungsnetzwerk 114 kann verschiedene Arten von Datenübertragungsnetzwerken enthalten, z.B. ein Weitverkehrsnetzwerk

(WAN), ein lokales Netzwerk (LAN), ein Telekommunikationsnetzwerk, ein drahtloses Netzwerk, ein öffentliches Wählnetz und/oder ein Satellitennetzwerk. Das Datenübertragungsnetzwerk 114 kann Verbindungen wie z.B. drahtgebundene und drahtlose Datenübertragungsverbindungen oder Lichtwellenleiterkabel enthalten. Dabei sollte offensichtlich sein, dass **Fig. 1** lediglich zur Veranschaulichung einer Realisierung dient und keinerlei Beschränkungen mit Blick auf die Umgebungen impliziert, in denen verschiedene Ausführungsformen realisiert werden können. Auf Grundlage von Entwurfs- und Realisierungsanforderungen können zahlreiche Änderungen an den dargestellten Umgebungen vorgenommen werden.

[0022] Gemäß einer Ausführungsform der Erfindung kann die Client-Datenverarbeitungseinheit 102 einen Prozessor 104 und eine Datenspeichereinheit 106 enthalten, die in der Lage ist, ein Software-Programm 108 und ein Programm 110A für eine automatische Sprache-zu-Text-Skalierung zu hosten und auszuführen sowie über das Datenübertragungsnetzwerk 114 mit dem Server 112 Daten auszutauschen. Die Client-Datenverarbeitungseinheit 102 kann zum Beispiel eine mobile Einheit, ein Telefon, ein persönlicher digitaler Assistent, ein Netbook, ein Laptop Computer, ein Desktop Computer oder jede Art von Datenverarbeitungseinheit sein, die in der Lage ist, ein Programm auszuführen und auf ein Netzwerk zuzugreifen. Wie unter Bezugnahme auf **Fig. 3** erläutert wird, kann die Client-Datenverarbeitungseinheit 102 interne Komponenten 302a bzw. externe Komponenten 304a enthalten.

[0023] Gemäß Ausführungsformen der Erfindung kann der Server-Computer 112 ein Laptop Computer, ein Netbook Computer, ein Personal Computer (PC), ein Desktop Computer bzw. jede programmierbare elektronische Einheit oder jedes Netzwerk von programmierbaren elektronischen Einheiten sein, die/das in der Lage ist, ein Programm 110B für eine automatische Sprache-zu-Text-Skalierung und eine Datenbank 116 zu hosten und über das Datenübertragungsnetzwerk 114 mit der Client-Datenverarbeitungseinheit 102 Daten auszutauschen. Wie unter Bezugnahme auf **Fig. 3** erläutert wird, kann der Server-Computer 112 interne Komponenten 302b bzw. externe Komponenten 304b enthalten. Der Server 112 kann außerdem in einem Cloud-Computing-Dienstmodell wie z.B. Software as a Service (SaaS), Platform as a Service (PaaS) oder Infrastructure as a Service (IaaS) betrieben werden. Der Server 112 kann sich auch in einem Cloud-Computing-Bereitstellungsmodell wie z.B. einer Private Cloud, einer Community Cloud, einer Public Cloud oder einer Hybrid Cloud befinden.

[0024] Gemäß der vorliegenden Ausführungsform kann das Programm 110A, 110B für eine automati-

sche Sprache-zu-Text-Skalierung in der Lage sein, für jedes Wort in einem Transkript ein Delta zwischen einer Prozesslaufzeit und einem Zeitpunkt zu berechnen, zu dem das Wort einem Client, d.h. einer Anwendungsschicht, bereitgestellt wird, die Deltas an eine Gruppe von Metrik-Servern zu übergeben, die konfiguriert werden, um die Deltas zu erfassen, von den Metrik-Servern aktuelle Werte der Deltas anzufordern, zu ermitteln, ob die aktuellen Werte der Deltas einen vordefinierten Schwellenwert überschreiten, und eine Zuordnung von Rechenressourcen auf Grundlage der aktuellen Werte der Deltas anzupassen, die den vordefinierten Schwellenwert überschreiten. Das Verfahren für eine automatische Sprache-zu-Text-Skalierung wird weiter unten in Bezug auf **Fig. 2** ausführlicher erläutert.

[0025] Unter Bezugnahme auf **Fig. 2** wird ein Betriebsablaufplan zum Zuordnen von Rechenressourcen in einem Prozess 200 für eine automatische Sprache-zu-Text-Skalierung gemäß mindestens einer Ausführungsform dargestellt. In einem Schritt 202 berechnet das Programm 110A, 110B für eine automatische Text-zu-Sprache-Skalierung das Delta für jedes Wort in dem Erkennungstranskript. Die berechneten Deltas können eine Zeitdifferenz zwischen der Prozesslaufzeit, wenn das Audiosignal für das Wort an die Spracherkennungs-Engine gesendet wird, und dem Zeitpunkt sein, zu dem das Wort dem Client, d.h. der Anwendungsschicht, bereitgestellt wird. Das Audiosignal kann in dem Moment als an die Spracherkennungs-Engine gesendet betrachtet werden, in dem das Wort durch den Benutzer ausgesprochen wird. Entsprechend kann das Audiosignal in dem Moment als an den Client gesendet betrachtet werden, indem das Wort transkribiert wird. Wenn ein Wort zum Beispiel um 13:00:00 Uhr ausgesprochen und mit einer Zeitmarke versehen wird und dasselbe Wort um 13:00:12 transkribiert und mit einer Zeitmarke versehen wird, kann das berechnete Delta als 12 Sekunden ermittelt werden. Die berechneten Deltas können ein umfassendes Latenzmaß sein, das eine Dienstgüte hinsichtlich einer Latenz ausdrückt, wie sie durch den Benutzer in einer bestimmten Audiositzung wahrgenommen wird.

[0026] Bei mindestens einer anderen Ausführungsform kann aus den aktuellen Werten der Deltas ein Histogramm erzeugt werden, um einen globalen Schätzwert einer Latenz zu erhalten. Somit kann die Aufwärts- oder Abwärtsskalierung der Rechenressourcen auf Grundlage eines Prozentsatzes von Datenpunkten erfolgen, die über dem vordefinierten Max-Latenz-Schwellenwert liegen. Wenn zum Beispiel 5 % der berechneten Deltas über X Sekunden liegen, kann das Programm 110A, 110B für eine automatische Text-zu-Sprache-Skalierung die Anzahl von Backend-Servern um Y Prozent nach oben skalieren. Der vordefinierte Max-Latenz-

Schwellenwert und der vordefinierte Min-Latenz-Schwellenwert können abhängig davon geändert werden, wie empfindlich Anwendungsfälle auf eine Latenz reagieren. In einer Audiositzung mit Übersprechen und Hintergrundgeräuschen kann der vordefinierte Max-Latenz-Schwellenwert abgesenkt werden, um mehr Rechenressourcen für diese bestimmte Sitzung vorzusehen.

[0027] Danach übergibt in einem Schritt 204 das Programm 110A, 110B für eine automatische Text-zu-Sprache-Skalierung die aktuellen Werte der Deltas an die Gruppe von Metrik-Servern, die konfiguriert werden, um die Deltas zu erfassen. Die aktuellen Werte der Deltas können über eine prozessübergreifende Datenübertragung übergeben werden, bei der das Programm 110A, 110B für eine automatische Text-zu-Sprache-Skalierung eine Anforderung sendet und die Gruppe von Metrik-Servern auf die Anforderung antwortet. So kann das Programm 110A, 110B für eine automatische Text-zu-Sprache-Skalierung zum Beispiel die aktuellen Werte der Deltas an die Metrik-Server senden, und die Metrik-Server können die Übergabe dieser aktuellen Werte akzeptieren.

[0028] In einem Schritt 206 veranlasst das Programm 110A, 110B für eine automatische Text-zu-Sprache-Skalierung dann die Gruppe von automatischen Horizontalskalierern, die aktuellen Werte der Deltas in regelmäßigen Abständen von den Metrik-Servern anzufordern. Die Anforderung kann zum Beispiel alle 10 Sekunden gestellt werden. Gemäß mindestens einer anderen Ausführungsform kann die Taktung der regelmäßig erfolgenden Anforderung geändert werden. So kann die Gruppe von automatischen Horizontalskalierern die Anforderung zum Beispiel alle 8 Sekunden oder alle 12 Sekunden stellen. Die Gruppe von automatischen Horizontalskalierern kann eine Metrik-Anwendungsprogrammierschnittstelle (Application Programming Interface, API) verwenden, um die aktuellen Werte der Deltas von den Metrik-Servern zu erhalten. Die Metrik-API kann ein Software-Vermittler sein, der analog zu der oben beschriebenen prozessübergreifenden Datenübertragung eine Anforderung sendet und eine Antwort bereitstellt. Die Gruppe von automatischen Horizontalskalierern kann zum Beispiel alle 10 Sekunden die aktuellen Werte der Deltas von den Metrik-Servern anfordern, und die API kann die aktuellen Werte wiederum der Gruppe von automatischen Horizontalskalierern bereitstellen.

[0029] Bei mindestens einer anderen Ausführungsform können die Metrik-Server konfiguriert werden, um die aktuellen Werte der Deltas ohne die Anforderung von der Gruppe von automatischen Horizontalskalierern an die Gruppe von automatischen Horizontalskalierern zu senden. Die Gruppe von Metrik-Servern kann die oben beschriebene Metrik-API ver-

wenden, um die aktuellen Werte der Deltas an die Gruppe von automatischen Horizontalskalierern zu senden. Zum Beispiel können die Metrik-Server die aktuellen Werte der Deltas an die Gruppe von automatischen Horizontalskalierern übergeben, und die Gruppe von automatischen Horizontalskalierern kann die Übergabe der aktuellen Werte der Deltas akzeptieren.

[0030] In einem Schritt 208 ermittelt das Programm 110A, 110B für eine automatische Text-zu-Sprache-Skalierung danach, ob die aktuellen Werte der Deltas den Schwellenwert für eine maximale Latenz (Max-Latenz-Schwellenwert) überschreiten. Die Ermittlung kann auf Grundlage einer Analyse erfolgen, wie viele Deltas den vordefinierten Max-Latenz-Schwellenwert überschreiten und wie viele Deltas unter dem vordefinierten Schwellenwert für eine minimale Latenz (Min-Latenz-Schwellenwert) liegen. Wenn das Programm 110A, 110B für eine automatische Text-zu-Sprache-Skalierung ermittelt, dass die aktuellen Werte der Deltas den vordefinierten Max-Latenz-Schwellenwert überschreiten, fährt der Prozess für eine automatische Text-zu-Sprache-Skalierung mit einem Schritt 210 fort, um eine Zuordnung von Rechenressourcen auf Grundlage der Häufigkeit der aktuellen Werte der Deltas anzupassen, die den vordefinierten Max-Latenz-Schwellenwert überschreiten.

[0031] Bei mindestens einer anderen Ausführungsform ermittelt das Programm 110A, 110B für eine automatische Text-zu-Sprache-Skalierung in einem Schritt 209, ob die aktuellen Werte der Deltas unter dem vordefinierten Min-Latenz-Schwellenwert liegen. Wenn das Programm 110A, 110B für eine automatische Text-zu-Sprache-Skalierung ermittelt, dass die aktuellen Werte der Deltas unter dem vordefinierten Min-Latenz-Schwellenwert liegen, fährt der Prozess für eine automatische Text-zu-Sprache-Skalierung mit einem Schritt 212 fort, um eine Zuordnung von Rechenressourcen auf Grundlage der Häufigkeit der aktuellen Werte der Deltas anzupassen, die unter dem vordefinierten Min-Latenz-Schwellenwert liegen.

[0032] Danach passt das Programm 110A, 110B für eine automatische Text-zu-Sprache-Skalierung in dem Schritt 210 die Zuordnung von Rechenressourcen auf Grundlage der Häufigkeit der aktuellen Werte der Deltas an, die den vordefinierten Max-Latenz-Schwellenwert überschreiten. Wenn der vordefinierte Max-Latenz-Schwellenwert zum Beispiel während mindestens Z Sekunden mehr als Y Mal überschritten wird, kann die Gruppe von automatischen Horizontalskalierern die Aufwärtsskalierungsoperation auslösen und der Spracherkennungs-Engine, die in dem entfernt angeordneten Rechenzentrum ausgeführt wird, weitere Rechenressourcen hinzufügen. Bevor die aktuellen Werte der Deltas verfügbar

sind, kann der Spracherkennungs-Engine anfänglich eine Standardanzahl von Rechenressourcen zugeordnet werden. Zum Beispiel kann zu Beginn der Audiositzung der Spracherkennungs-Engine mindestens eine Rechenressource zugeordnet werden. Die Gruppe von automatischen Horizontalskalierern kann die Aufwärtsskalierungsoperation auslösen, indem ein Signal an eine Überwachungsschaltung innerhalb der Backend-Server gesendet wird, um die Backend-Server hochzufahren. Die Rechenressourcen können inkrementell so lange hinzugefügt werden, bis die aktuellen Werte der Deltas innerhalb des akzeptablen Bereichs für eine Latenz liegen.

[0033] Bei mindestens einer anderen Ausführungsform kann eine maximale Anzahl von Rechenressourcen der Spracherkennungs-Engine zugeordnet werden, um eine Überbuchung der Rechenressourcen für eine einzelne Audiositzung zu vermeiden. Das Programm 110A, 110B für eine automatische Text-zu-Sprache-Skalierung kann konfiguriert werden, um neue Audiositzungsanforderungen von den Benutzern abzuweisen und so die Dienstgüte der bereits hergestellten Audiositzungen zu garantieren. Wenn zum Beispiel zum aktuellen Zeitpunkt der Gesamtheit aller Audiositzungen 90 % der verfügbaren Rechenressourcen zugeordnet werden, kann das Programm 110A, 110B für eine automatische Text-zu-Sprache-Skalierung neue Audiositzungsanforderungen abweisen.

[0034] Bei mindestens einer anderen Ausführungsform passt das Programm 110A, 110B für eine automatische Text-zu-Sprache-Skalierung in einem Schritt 212 die Zuordnung von Rechenressourcen auf Grundlage der Häufigkeit der aktuellen Werte der Deltas an, die unter dem vordefinierten Schwellenwert für eine minimale Latenz (Min-Latenz-Schwellenwert) liegen. Wenn der vordefinierte Min-Latenz-Schwellenwert zum Beispiel während mindestens Z Sekunden nicht mehr als Y Mal überschritten wird, kann die Gruppe von automatischen Horizontalskalierern die Abwärtsskalierungsoperation auslösen und die Rechenressourcen reduzieren, die der Spracherkennungs-Engine zugeordnet werden, die in dem entfernt angeordneten Rechenzentrum ausgeführt wird. Die Gruppe von automatischen Horizontalskalierern kann die Abwärtsskalierungsoperation auslösen, indem ein Signal an die Überwachungsschaltung innerhalb der Backend-Server gesendet wird, um die Backend-Server herunterzufahren. Entsprechend können die Rechenressourcen inkrementell so lange reduziert werden, bis die aktuellen Werte der Deltas innerhalb des akzeptablen Bereichs für eine Latenz liegen.

[0035] Bei mindestens einer anderen Ausführungsform kann eine minimale Anzahl von Rechenressourcen der Spracherkennungs-Engine zugeordnet werden, damit die Dienstgüte der Audiositzungen nicht

signifikant beeinträchtigt wird. Zum Beispiel können zu jedem beliebigen Zeitpunkt mindestens 10 % der verfügbaren Rechenressourcen der Spracherkennungs-Engine zugeordnet werden. Selbst wenn die aktuellen Werte der Deltas innerhalb des akzeptablen Bereichs für eine Latenz liegen, kann somit für jede Audiositzung eine Anzahl von Rechenressourcen vorgesehen werden, um die Dienstgüte zu garantieren.

[0036] Dabei sollte klar sein, dass **Fig. 2** lediglich zur Veranschaulichung einer Realisierung dient und keinerlei Beschränkungen mit Blick auf die Art und Weise impliziert, wie verschiedene Ausführungsformen realisiert werden können. Auf Grundlage von Entwurfs- und Realisierungsanforderungen können zahlreiche Änderungen an den dargestellten Umgebungen vorgenommen werden.

[0037] **Fig. 3** ist ein Blockschaubild 300 von internen und externen Komponenten der in **Fig. 1** abgebildeten Datenverarbeitungseinheit 112 und des Servers 112 gemäß einer Ausführungsform der vorliegenden Erfindung. Dabei sollte klar sein, dass **Fig. 3** lediglich zur Veranschaulichung einer Realisierung dient und keinerlei Beschränkungen mit Blick auf die Umgebungen impliziert, in denen verschiedene Ausführungsformen realisiert werden können. Auf Grundlage von Entwurfs- und Realisierungsanforderungen können zahlreiche Änderungen an den dargestellten Umgebungen vorgenommen werden.

[0038] Das Datenverarbeitungssystem 302, 304 ist repräsentativ für jede beliebige elektronische Einheit, die in der Lage ist, maschinenlesbare Programmweisungen auszuführen. Das Datenverarbeitungssystem 302, 304 kann repräsentativ für ein Smartphone, ein Computersystem, einen PDA oder andere elektronische Einheiten sein. Beispiele von Datenverarbeitungssystemen, -umgebungen und/oder -konfigurationen, die durch das Datenverarbeitungssystem 302, 304 repräsentiert werden können, sind, ohne darauf beschränkt zu sein, Personal-Computersysteme, Server-Computersysteme, Thin Clients, Thick Clients, Handheld- oder Laptop-Einheiten, Mehrprozessorsysteme, Systeme auf Grundlage von Mikroprozessoren, Netzwerk-PCs, Mini-Computersysteme, Mainframe-Computersysteme sowie verteilte Cloud-Computing-Umgebungen, die eines bzw. eine der obigen Systeme oder Einheiten enthalten.

[0039] Die Client-Datenverarbeitungseinheit 102 und der Server 112 können jeweils Sätze der internen Komponenten 302 a,b und der externen Komponenten 304 a,b enthalten, die in **Fig. 3** veranschaulicht werden. Jeder der Sätze der internen Komponenten 302 enthält einen oder mehrere Prozessoren 320, einen oder mehrere computerlesbare RAMs 322 und einen oder mehrere computerlesbare

ROMs 324 auf einem oder mehreren Bussen 326 sowie ein oder mehrere Betriebssysteme 328 und eine oder mehrere computerlesbare physische Speichereinheiten 330. Das eine oder die mehreren Betriebssysteme 328, das Software-Programm 108 und das Programm 110A für eine automatische Sprache-zu-Text-Skalierung in der Client-Datenverarbeitungseinheit 102 sowie das Programm 110B für eine automatische Sprache-zu-Text-Skalierung auf dem Server 112 werden auf einer oder mehreren der computerlesbaren physischen Speichereinheiten 330 gespeichert, um über einen oder mehrere der betreffenden RAMs 322 (die üblicherweise einen Cache-Arbeitsspeicher enthalten) durch einen oder mehrere der betreffenden Prozessoren 320 ausgeführt zu werden. Bei der in **Fig. 3** veranschaulichten Ausführungsform ist jede der computerlesbaren physischen Speichereinheiten 330 eine Magnetplattenspeichereinheit einer internen Festplatte. Alternativ ist jede der computerlesbaren physischen Speichereinheiten 330 eine Halbleiterspeichereinheit wie z.B. ein ROM 324, ein EPROM, ein Flash-Arbeitsspeicher oder eine beliebige andere computerlesbare physische Speichereinheit, die ein Computerprogramm und digitale Daten speichern kann.

[0040] Jeder Satz der internen Komponenten 302 a,b enthält außerdem ein Lese/Schreib-Laufwerk oder eine Lese/Schreib-Schnittstelle 332, um von einer oder mehreren tragbaren, computerlesbaren physischen Speichereinheiten 338 wie z.B. einem CD-ROM, einer DVD, einem Speicher-Stick, einem Magnetband, einer Magnetplatte, einer optischen Platte oder einer Halbleiterspeichereinheit lesen und darauf schreiben zu können. Ein Software-Programm wie z.B. das Programm 110A, 110B für eine automatische Sprache-zu-Text-Skalierung kann auf einer oder mehreren der betreffenden tragbaren, computerlesbaren physischen Speichereinheiten 338 gespeichert, über das betreffende Lese/Schreib-Laufwerk oder die betreffende Lese/Schreib-Schnittstelle 332 gelesen und auf die betreffende Festplatte 330 geladen werden.

[0041] Jeder Satz der internen Komponenten 302 a,b enthält außerdem Netzwerkadapter oder -schnittstellen 336 wie z.B. TCP/IP-Adapterkarten, WLAN-Funkschnittstellenkarten oder 3G- bzw. 4G-Funkschnittstellenkarten oder andere drahtgebundene oder drahtlose Datenübertragungsverbindungen. Das Software-Programm 108 und das Programm 110A für eine automatische Text-zu-Sprache-Skalierung in der Client-Datenverarbeitungseinheit 102 sowie das Programm 110B für eine automatische Text-zu-Sprache-Skalierung auf dem Server 112 können über ein Netzwerk (zum Beispiel das Internet, ein lokales Netzwerk oder aber ein Weitverkehrsnetzwerk) und die betreffenden Netzwerkadapter oder -schnittstellen 336 von einem externen Computer auf die Client-Datenverarbeitungseinheit

102 und den Server 112 heruntergeladen werden. Von den Netzwerkadaptern oder -schnittstellen 336 werden das Software-Programm 108 und das Programm 110A für eine automatische Text-zu-Sprache-Skalierung in der Client-Datenverarbeitungseinheit 102 sowie das Programm 110B für eine automatische Text-zu-Sprache-Skalierung auf dem Server 112 auf die betreffende Festplatte 330 geladen. Das Netzwerk kann Kupferkabel, Lichtwellenleiter, eine drahtlose Übertragung, Router, Firewalls, Switches, Gateway Computer und/oder Edge-Server aufweisen.

[0042] Jeder der Sätze der externen Komponenten 304 a,b kann einen Computer-Anzeigebildschirm 344, eine Tastatur 342 und eine Computermouse 334 enthalten. Die externen Komponenten 304 a,b können auch berührungsempfindliche Bildschirme, virtuelle Tastaturen, Tastfelder, Zeigeeinheiten und andere Benutzerinteraktionseinheiten enthalten. Jeder der Sätze der internen Komponenten 302 a,b enthält außerdem Einheitentreiber 340, um eine Schnittstelle mit dem Computer-Anzeigebildschirm 344, der Tastatur 342 und der Computermouse 334 bereitzustellen. Die Einheitentreiber 340, das Lese/Schreib-Laufwerk oder die Lese/Schreib-Schnittstelle 332 und der Netzwerkadapter oder die Netzwerkschnittstelle 336 weisen Hardware und Software auf (die in der Speichereinheit 330 und/oder dem ROM 324 gespeichert werden).

[0043] Es sei von vornherein klargestellt, dass das Umsetzen der hierin angeführten Lehren nicht auf eine Cloud-Computing-Umgebung beschränkt ist, obwohl diese Offenbarung eine ausführliche Beschreibung von Cloud Computing enthält. Vielmehr können Ausführungsformen der vorliegenden Erfindung gemeinsam mit jeder beliebigen Art von jetzt bekannter oder später entwickelter Datenverarbeitungsumgebung umgesetzt werden.

[0044] Cloud Computing ist ein Dienstbereitstellungsmodell zum Ermöglichen eines problemlosen bedarfsgesteuerten Netzwerkzugriffs auf einen gemeinsam genutzten Pool von konfigurierbaren Datenverarbeitungsressourcen (z.B. Netzwerke, Netzwerkbandbreite, Server, Verarbeitung, Hauptspeicher, Speicher, Anwendungen, virtuelle Maschinen und Dienste), die mit minimalem Verwaltungsaufwand bzw. minimaler Interaktion mit einem Anbieter des Dienstes schnell bereitgestellt und freigegeben werden können. Dieses Cloud-Modell kann mindestens fünf Eigenschaften, mindestens drei Dienstmodelle und mindestens vier Einsatzmodelle enthalten.

[0045] Bei den Eigenschaften handelt es sich um die Folgenden:

On-Demand Self-Service: Ein Cloud-Nutzer kann einseitig automatisch nach Bedarf für

Datenverarbeitungsfähigkeiten wie Server-Zeit und Netzwerkspeicher sorgen, ohne dass eine menschliche Interaktion mit dem Anbieter der Dienste erforderlich ist.

Broad Network Access: Es sind Fähigkeiten über ein Netzwerk verfügbar, auf die durch Standardmechanismen zugegriffen wird, die die Verwendung durch heterogene Thin- oder Thick-Client-Plattformen (z.B. Mobiltelefone, Laptops und PDAs) unterstützen.

Resource Pooling: Die Datenverarbeitungsressourcen des Anbieters werden zusammengefasst, um mehreren Nutzern unter Verwendung eines Multi-Tenant-Modells zu dienen, wobei verschiedene physische und virtuelle Ressourcen dynamisch nach Bedarf zugewiesen und neu zugewiesen werden. Es gibt eine gefühlte Standortunabhängigkeit, da der Nutzer allgemein keine Kontrolle bzw. Kenntnis über den genauen Standort der bereitgestellten Ressourcen hat, aber in der Lage sein kann, einen Standort auf einer höheren Abstraktionsebene festzulegen (z.B. Land, Staat oder Rechenzentrum).

Rapid Elasticity: Funktionen können für eine schnelle horizontale Skalierung (Scale-out) schnell und elastisch bereitgestellt werden, in einigen Fällen auch automatisch, und für ein schnelles Scale-in schnell freigegeben werden. Für den Nutzer erscheinen die für das Bereitstellen verfügbaren Funktionen häufig unbegrenzt und sie können jederzeit in jeder beliebigen Menge gekauft werden.

Measured Service: Cloud-Systeme steuern und optimieren die Verwendung von Ressourcen automatisch, indem sie eine Messfunktion auf einer gewissen Abstraktionsebene nutzen, die für die Art von Dienst geeignet ist (z.B. Speicher, Verarbeitung, Bandbreite sowie aktive Benutzerkonten). Der Ressourcen-Verbrauch kann überwacht, gesteuert und gemeldet werden wodurch sowohl für den Anbieter als auch für den Nutzer des verwendeten Dienstes Transparenz geschaffen wird.

[0046] Bei den Dienstmodellen handelt es sich um die Folgenden:

Software as a Service (SaaS): Die dem Nutzer bereitgestellte Funktion besteht darin, die in einer Cloud-Infrastruktur laufenden Anwendungen des Anbieters zu verwenden. Die Anwendungen sind über eine Thin-Client-Schnittstelle wie einen Web-Browser (z.B. auf dem Web beruhende E-Mail) von verschiedenen Client-Einheiten her zugänglich. Der Nutzer verwaltet bzw. steuert die zugrunde liegende Cloud-Infrastruktur nicht, darunter das Netzwerk, Server, Betriebssysteme, Speicher bzw. sogar einzelne

Anwendungsfähigkeiten, mit der möglichen Ausnahme von eingeschränkten nutzerspezifischen Anwendungskonfigurationseinstellungen.

Platform as a Service (PaaS): Die dem Nutzer bereitgestellte Funktion besteht darin, durch einen Nutzer erstellte bzw. erhaltene Anwendungen, die unter Verwendung von durch den Anbieter unterstützten Programmiersprachen und Tools erstellt wurden, in der Cloud-Infrastruktur einzusetzen. Der Nutzer verwaltet bzw. steuert die zugrunde liegende Cloud-Infrastruktur nicht, darunter Netzwerke, Server, Betriebssysteme bzw. Speicher, hat aber die Kontrolle über die eingesetzten Anwendungen und möglicherweise über Konfigurationen für Application Hosting Environment.

Infrastructure as a Service (IaaS): Die dem Nutzer bereitgestellte Funktion besteht darin, das Verarbeiten, Speichern, Netzwerke und andere grundlegende Datenverarbeitungsressourcen bereitzustellen, wobei der Nutzer in der Lage ist, beliebige Software einzusetzen und auszuführen, zu der Betriebssysteme und Anwendungen gehören können. Der Nutzer verwaltet bzw. steuert die zugrunde liegende Cloud-Infrastruktur nicht, hat aber die Kontrolle über Betriebssysteme, Speicher, eingesetzte Anwendungen und möglicherweise eine eingeschränkte Kontrolle über ausgewählte Netzwerkkomponenten (z.B. Host-Firewalls).

[0047] Bei den Einsatzmodellen handelt es sich um die Folgenden:

Private Cloud: Die Cloud-Infrastruktur wird einzig und allein für eine Organisation betrieben. Sie kann durch die Organisation oder einen Dritten verwaltet werden und kann sich in den eigenen Räumen oder in fremden Räumen befinden.

Community Cloud: Die Cloud-Infrastruktur wird von mehreren Organisationen gemeinsam genutzt und unterstützt eine spezielle Benutzergemeinschaft, die gemeinsame Angelegenheiten hat (z.B. Zielsetzung, Sicherheitsanforderungen, Richtlinien sowie Überlegungen bezüglich der Einhaltung von Vorschriften). Sie kann durch die Organisationen oder einen Dritten verwaltet werden und sich in den eigenen Räumen oder in fremden Räumen befinden.

Public Cloud: Die Cloud-Infrastruktur wird der allgemeinen Öffentlichkeit oder einer großen Industriegruppe zur Verfügung gestellt und gehört einer Cloud-Dienste verkaufenden Organisation.

Hybrid Cloud: Die Cloud-Infrastruktur ist eine Zusammensetzung aus zwei oder mehreren Clouds (privat, Nutzergemeinschaft oder öffent-

lich), die zwar einzelne Einheiten bleiben, aber durch eine standardisierte oder proprietäre Technologie miteinander verbunden sind, die Daten- und Anwendungsportierbarkeit ermöglicht (z.B. Cloud-Zielgruppenverteilung für den Lastenausgleich zwischen Clouds).

[0048] Eine Cloud-Computing-Umgebung ist dienstorientiert mit einem Fokus auf Statusunabhängigkeit, geringer Kopplung, Modularität und semantischer Interoperabilität. Im Mittelpunkt des Cloud Computing steht eine Infrastruktur, die ein Netzwerk aus zusammengeschalteten Knoten aufweist.

[0049] Unter Bezugnahme auf **Fig. 4** wird eine veranschaulichende Cloud-Computing-Umgebung 40 dargestellt. Wie gezeigt, weist die Cloud-Computing-Umgebung 40 einen oder mehrere Cloud-Computing-Knoten 100 auf, mit denen von Cloud-Nutzern verwendete lokale Datenverarbeitungseinheiten wie der elektronische Assistent (PDA) oder das Mobiltelefon 44A, der Desktop Computer 44B, der Laptop Computer 44C und/oder das Automobil-Computersystem 44N Daten austauschen können. Die Knoten 100 können miteinander Daten austauschen. Sie können physisch oder virtuell in einem oder mehreren Netzwerken wie Private, Community, Public oder Hybrid Clouds zusammengefasst werden (nicht gezeigt), wie vorstehend beschrieben wurde, oder aber in einer Kombination daraus. Dies ermöglicht es der Cloud-Computing-Umgebung 40, 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. 4** gezeigten Datenverarbeitungseinheiten 44A bis N lediglich veranschaulichend sein sollen und dass die Datenverarbeitungsknoten 100 und die Cloud-Computing-Umgebung 40 über eine beliebige Art von 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.

[0050] Unter Bezugnahme auf **Fig. 5** wird ein Satz von funktionalen Abstraktionsschichten 500 gezeigt, die durch die Cloud-Computing-Umgebung 40 bereitgestellt werden. Es sollte von vornherein klar sein, dass die in **Fig. 5** gezeigten Komponenten, Schichten und Funktionen lediglich veranschaulichend sein sollen und Ausführungsformen der Erfindung nicht darauf beschränkt sind. Wie dargestellt, werden die folgenden Schichten und entsprechenden Funktionen bereitgestellt:

Eine Hardware- und Software-Schicht 60 enthält Hardware- und Software-Komponenten. Zu Beispielen für Hardware-Komponenten gehören: Mainframe Computer 61; auf der RISC(Reduced Instruction Set Computer)-

Architektur beruhende Server 62; Server 63; Blade-Server 64; Speichereinheiten 65; und Netzwerke sowie Netzwerkkomponenten 66. Bei einigen Ausführungsformen enthalten Software-Komponenten eine Netzwerk-Anwendungsserver-Software 67 und eine Datenbank-Software 68.

[0051] 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.

[0052] 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. Eine Sicherheit stellt die Identitätsüberprüfung für Cloud-Nutzer und Aufgaben sowie Schutz für Daten und andere Ressourcen bereit. Ein Benutzerportal 83 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, sodass die benötigten Dienstziele erreicht werden. Ein Planen und Erfüllen von Vereinbarungen zum Dienstumfang (Service Level Agreement, SLA) 85 stellt die Vorab-Anordnung und die Beschaffung von Cloud-Computing-Ressourcen, für die eine zukünftige Anforderung vorausgesehen wird, gemäß einer SLA bereit.

[0053] Eine Arbeitslastenschicht 90 stellt Beispiele für die Funktionalität bereit, für welche die Cloud-Computing-Umgebung verwendet werden kann. Beispiele für Auslastungen und Funktionen, die von dieser Schicht bereitgestellt werden können, lauten: Zuordnung und Navigation 91; Software-Entwicklung und Lebenszyklusverwaltung 92; Bereitstellung von virtuellen Schulungen 93; Datenanalyseverarbeitung 94; Transaktionsverarbeitung 95; und eine automatische Sprache-zu-Text-Skalierung durch eine Beschreibung in natürlicher Sprache 96. Die automatische Sprache-zu-Text-Skalierung durch eine Beschreibung in natürlicher Sprache 96 kann sich auf ein dynamisches Erzeugen von Metriken beziehen, die mit der Benutzererfahrung und den Rechen-

ressourcen korrelieren, die zu einem beliebigen Zeitpunkt benötigt werden, und entsprechend die Rechenressourcen zuzuordnen, die notwendig sind, um eine Latenz innerhalb eines akzeptablen Bereichs zu halten.

[0054] Die Beschreibungen der verschiedenen Ausführungsformen der vorliegenden Erfindung wurden zum Zwecke der Veranschaulichung vorgelegt und sind nicht als vollständig oder auf die offenbarten Ausführungsformen beschränkt zu verstehen. Der Fachmann weiß, dass zahlreiche Modifikationen und Abwandlungen möglich sind, ohne vom inhaltlichen Umfang der beschriebenen Ausführungsformen abzuweichen. Die hierin verwendete Begrifflichkeit wurde gewählt, um die Grundsätze der Ausführungsformen, die praktische Anwendung oder technische Verbesserung gegenüber markt gängigen Technologien bestmöglich zu erläutern bzw. anderen Fachleuten das Verständnis der hierin offenbarten Ausführungsformen zu ermöglichen.

Patentansprüche

1. Computergestütztes Verfahren zum Zuordnen von Rechenressourcen, wobei das Verfahren aufweist:
Berechnen eines Deltas zwischen einer Prozesslaufzeit und einem Zeitpunkt, zu dem das Wort einem Client bereitgestellt wird, für jedes Wort in einem Transkript;
Übergeben der Deltas an eine Gruppe von Metrik-Servern, wobei die Gruppe von Metrik-Servern konfiguriert wird, um die Deltas zu erfassen;
Anfordern von aktuellen Werten der Deltas von der Gruppe von Metrik-Servern;
Ermitteln, ob die aktuellen Werte der Deltas einen vordefinierten Schwellenwert für eine maximale Latenz, Max-Latenz-Schwellenwert, überschreiten;
und
als Reaktion auf ein Ermitteln, dass die aktuellen Werte der Deltas den vordefinierten Max-Latenz-Schwellenwert überschreiten, Anpassen der zugeordneten Rechenressourcen auf Grundlage einer Häufigkeit der aktuellen Werte der Deltas, die den vordefinierten Max-Latenz-Schwellenwert überschreiten.
2. Verfahren nach Anspruch 1, wobei der vordefinierte Max-Latenz-Schwellenwert konfiguriert wird, um abhängig davon geändert zu werden, wie empfindlich Anwendungsfälle auf eine Latenz reagieren.
3. Verfahren nach Anspruch 1, wobei die zugeordneten Rechenressourcen inkrementell so lange hinzugefügt werden, bis die aktuellen Werte der Deltas innerhalb des akzeptablen Bereichs für eine Latenz liegen.
4. Verfahren nach Anspruch 1, wobei ein Anpassen der zugeordneten Rechenressourcen auf Grundlage der Häufigkeit der aktuellen Werte der Deltas erfolgt, die unter einem vordefinierten Schwellenwert für eine minimale Latenz, Min-Latenz-Schwellenwert, liegen.
5. Verfahren nach Anspruch 4, wobei der vordefinierte Min-Latenz-Schwellenwert konfiguriert wird, um abhängig davon geändert zu werden, wie empfindlich Anwendungsfälle auf eine Latenz reagieren.
6. Verfahren nach Anspruch 1, wobei die zugeordneten Rechenressourcen inkrementell so lange reduziert werden, bis die aktuellen Werte der Deltas innerhalb des akzeptablen Bereichs für eine Latenz liegen.
7. Verfahren nach Anspruch 1, wobei ein Berechnen des Deltas zwischen der Prozesslaufzeit und dem Zeitpunkt, zu dem das Wort dem Client bereitgestellt wird, für jedes Wort in dem Transkript des Weiteren aufweist:
Erzeugen eines Histogramms aus den aktuellen Werten der Deltas; und
Durchführen einer Aufwärtsskalierung der zugeordneten Rechenressourcen auf Grundlage eines Prozentsatzes von Datenpunkten, die über dem vordefinierten Max-Latenz-Schwellenwert liegen.
8. Computersystem, wobei das Computersystem aufweist:
einen oder mehrere Prozessoren (320), einen oder mehrere computerlesbare Arbeitsspeicher, ein oder mehrere computerlesbare physische Speichermedien und Programmanweisungen, die auf mindestens einem des einen oder der mehreren physischen Speichermedien gespeichert werden, um durch mindestens einen des einen oder der mehreren Prozessoren (320) über mindestens einen des einen oder der mehreren Arbeitsspeicher ausgeführt zu werden, wobei das Computersystem zum Durchführen eines Verfahrens in der Lage ist, aufweisend:
Berechnen eines Deltas zwischen einer Prozesslaufzeit und einem Zeitpunkt, zu dem das Wort einem Client bereitgestellt wird, für jedes Wort in einem Transkript;
Übergeben der Deltas an eine Gruppe von Metrik-Servern, wobei die Gruppe von Metrik-Servern konfiguriert wird, um die Deltas zu erfassen;
Anfordern von aktuellen Werten der Deltas von der Gruppe von Metrik-Servern;
Ermitteln, ob die aktuellen Werte der Deltas einen vordefinierten Max-Latenz-Schwellenwert überschreiten; und
als Reaktion auf ein Ermitteln, dass die aktuellen Werte der Deltas den vordefinierten Max-Latenz-Schwellenwert überschreiten, Anpassen der zugeordneten Rechenressourcen auf Grundlage einer Häufigkeit der aktuellen Werte der Deltas, die den

vordefinierten Max-Latenz-Schwellenwert überschreiten.

9. Computersystem nach Anspruch 8, wobei der vordefinierte Max-Latenz-Schwellenwert konfiguriert wird, um abhängig davon geändert zu werden, wie empfindlich Anwendungsfälle auf eine Latenz reagieren.

10. Computersystem nach Anspruch 8, wobei die zugeordneten Rechenressourcen inkrementell so lange hinzugefügt werden, bis die aktuellen Werte der Deltas innerhalb des akzeptablen Bereichs für eine Latenz liegen.

11. Computersystem nach Anspruch 8, wobei ein Anpassen der zugeordneten Rechenressourcen auf Grundlage der Häufigkeit der aktuellen Werte der Deltas erfolgt, die unter einem vordefinierten Min-Latenz-Schwellenwert liegen.

12. Computersystem nach Anspruch 11, wobei der vordefinierte Min-Latenz-Schwellenwert konfiguriert wird, um abhängig davon geändert zu werden, wie empfindlich Anwendungsfälle auf eine Latenz reagieren.

13. Computersystem nach Anspruch 8, wobei die zugeordneten Rechenressourcen inkrementell so lange reduziert werden, bis die aktuellen Werte der Deltas innerhalb des akzeptablen Bereichs für eine Latenz liegen.

14. Computersystem nach Anspruch 8, wobei ein Berechnen des Deltas zwischen der Prozesslaufzeit und dem Zeitpunkt, zu dem das Wort dem Client bereitgestellt wird, für jedes Wort in dem Transkript des Weiteren aufweist:
Erzeugen eines Histogramms aus den aktuellen Werten der Deltas; und
Durchführen einer Aufwärtsskalierung der zugeordneten Rechenressourcen auf Grundlage eines Prozentsatzes von Datenpunkten, die über dem vordefinierten Max-Latenz-Schwellenwert liegen.

15. Computerprogrammprodukt, wobei das Computerprogrammprodukt aufweist:
ein oder mehrere computerlesbare physische Speichermedien und Programmanweisungen, die auf mindestens einem des einen oder der mehreren physischen Speichermedien gespeichert werden, wobei die Programmanweisungen durch einen Prozessor (320) ausführbar sind, der zum Durchführen eines Verfahrens in der Lage ist, wobei das Verfahren aufweist:
Berechnen eines Deltas zwischen einer Prozesslaufzeit und einem Zeitpunkt, zu dem das Wort einem Client bereitgestellt wird, für jedes Wort in einem Transkript;
Übergeben der Deltas an eine Gruppe von Metrik-

Servern, wobei die Gruppe von Metrik-Servern konfiguriert wird, um die Deltas zu erfassen;
Anfordern von aktuellen Werten der Deltas von der Gruppe von Metrik-Servern;
Ermitteln, ob die aktuellen Werte der Deltas einen vordefinierten Max-Latenz-Schwellenwert überschreiten; und
als Reaktion auf ein Ermitteln, dass die aktuellen Werte der Deltas den vordefinierten Max-Latenz-Schwellenwert überschreiten, Anpassen der zugeordneten Rechenressourcen auf Grundlage einer Häufigkeit der aktuellen Werte der Deltas, die den vordefinierten Max-Latenz-Schwellenwert überschreiten.

16. Computerprogrammprodukt nach Anspruch 15, wobei der vordefinierte Max-Latenz-Schwellenwert konfiguriert wird, um abhängig davon geändert zu werden, wie empfindlich Anwendungsfälle auf eine Latenz reagieren.

17. Computerprogrammprodukt nach Anspruch 15, wobei die zugeordneten Rechenressourcen inkrementell so lange hinzugefügt werden, bis die aktuellen Werte der Deltas innerhalb des akzeptablen Bereichs für eine Latenz liegen.

18. Computerprogrammprodukt nach Anspruch 15, wobei ein Anpassen der zugeordneten Rechenressourcen auf Grundlage der Häufigkeit der aktuellen Werte der Deltas erfolgt, die unter einem vordefinierten Min-Latenz-Schwellenwert liegen.

19. Computerprogrammprodukt nach Anspruch 18, wobei der vordefinierte Min-Latenz-Schwellenwert konfiguriert wird, um abhängig davon geändert zu werden, wie empfindlich Anwendungsfälle auf eine Latenz reagieren.

20. Computerprogrammprodukt nach Anspruch 15, wobei die zugeordneten Rechenressourcen inkrementell so lange reduziert werden, bis die aktuellen Werte der Deltas innerhalb des akzeptablen Bereichs für eine Latenz liegen.

Es folgen 5 Seiten Zeichnungen

Anhängende Zeichnungen

100 ↘

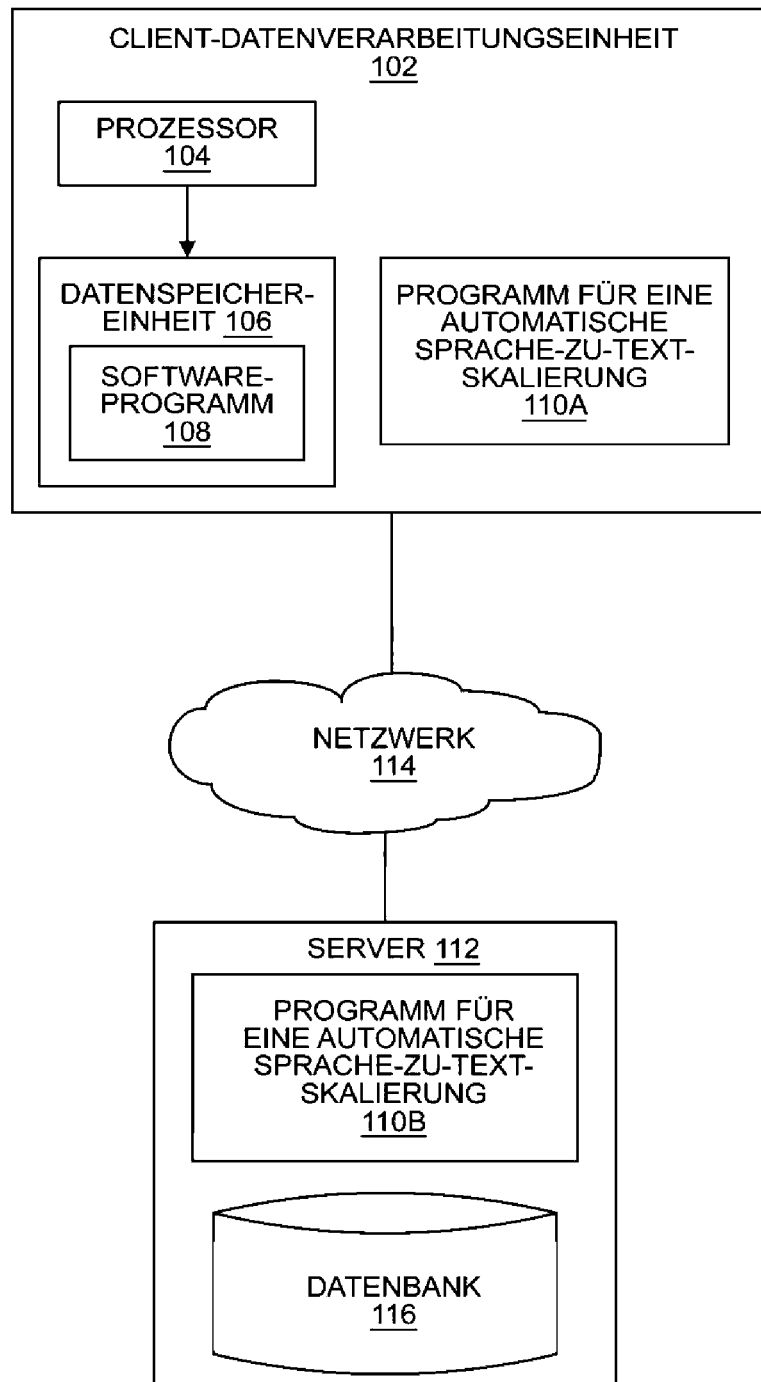


FIG. 1

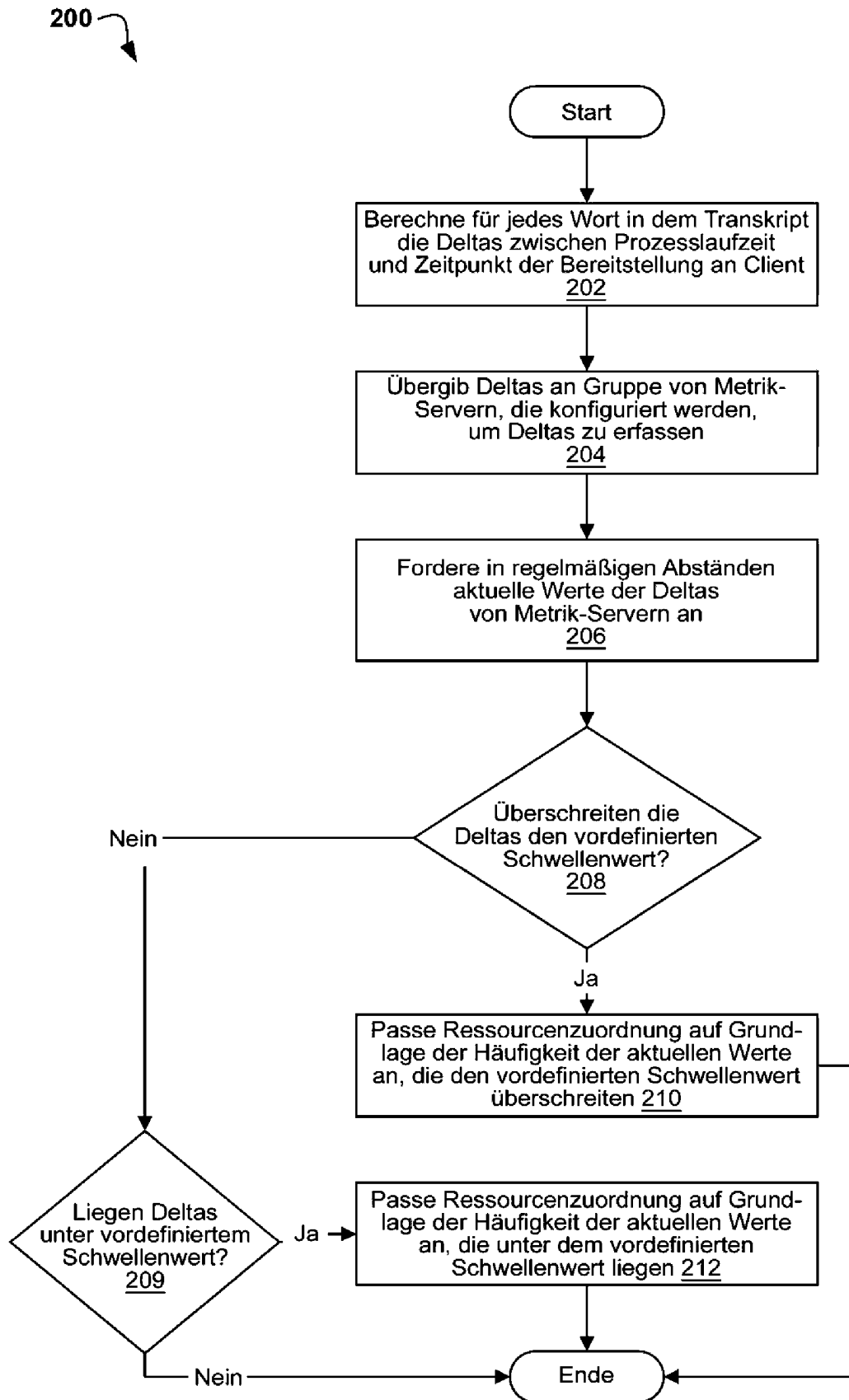


FIG. 2

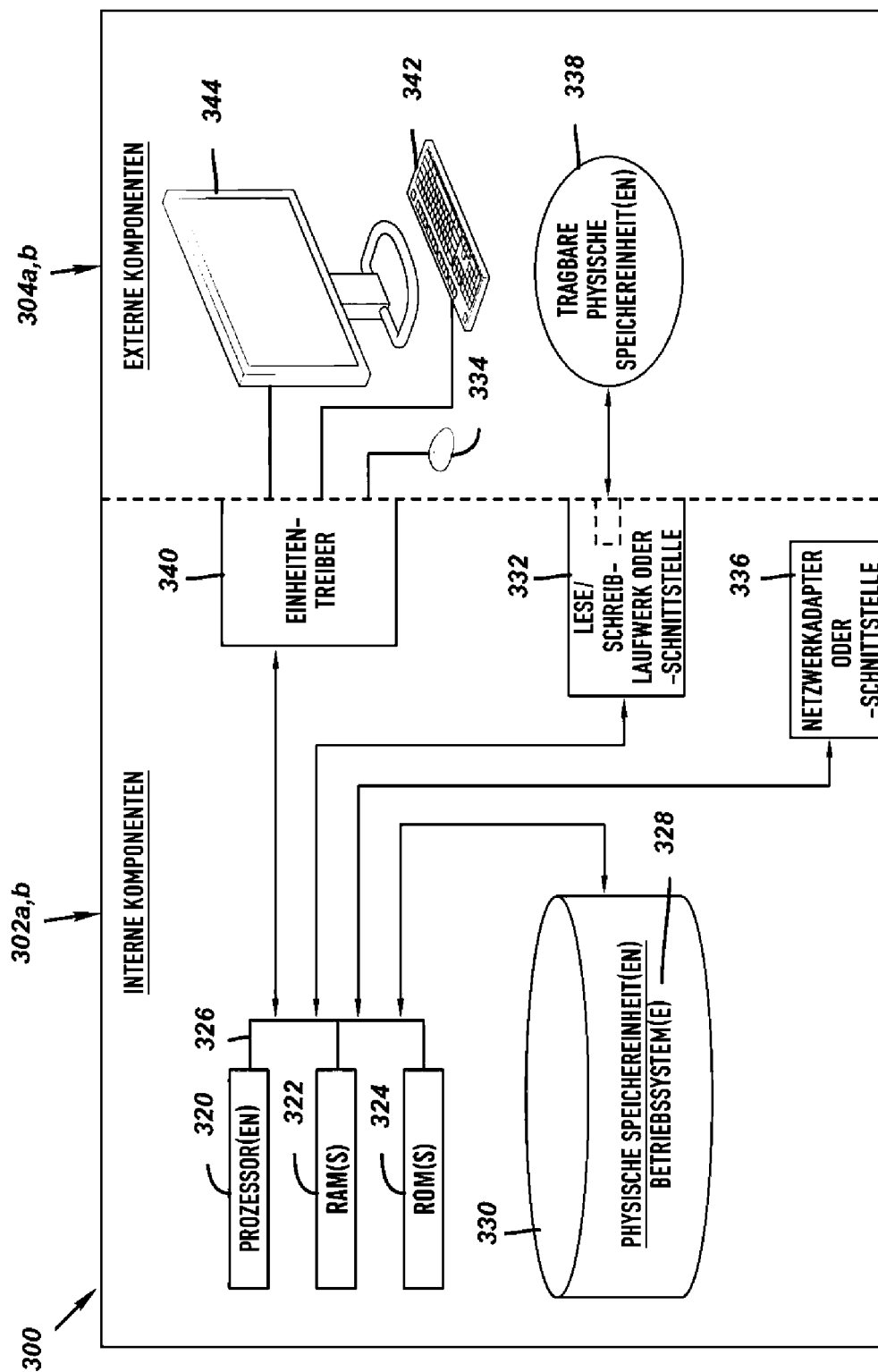


FIG. 3

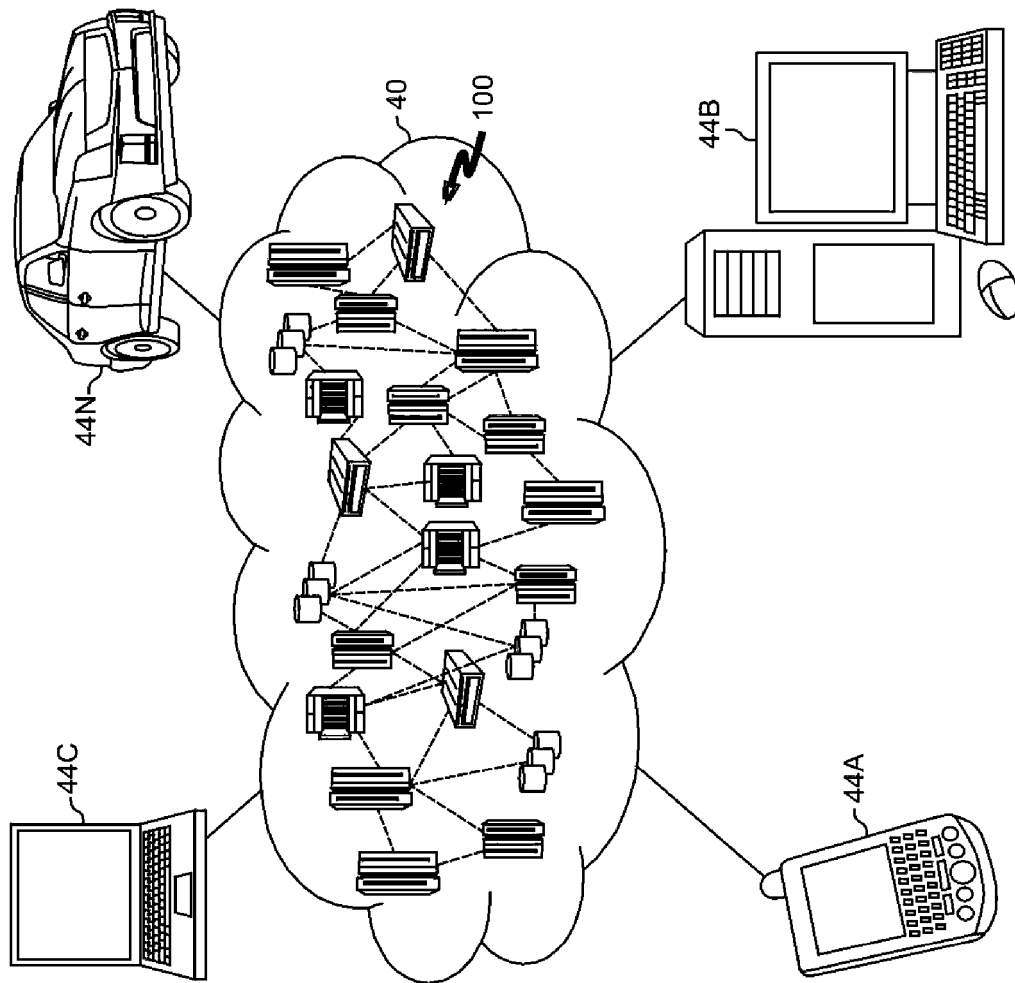


FIG. 4

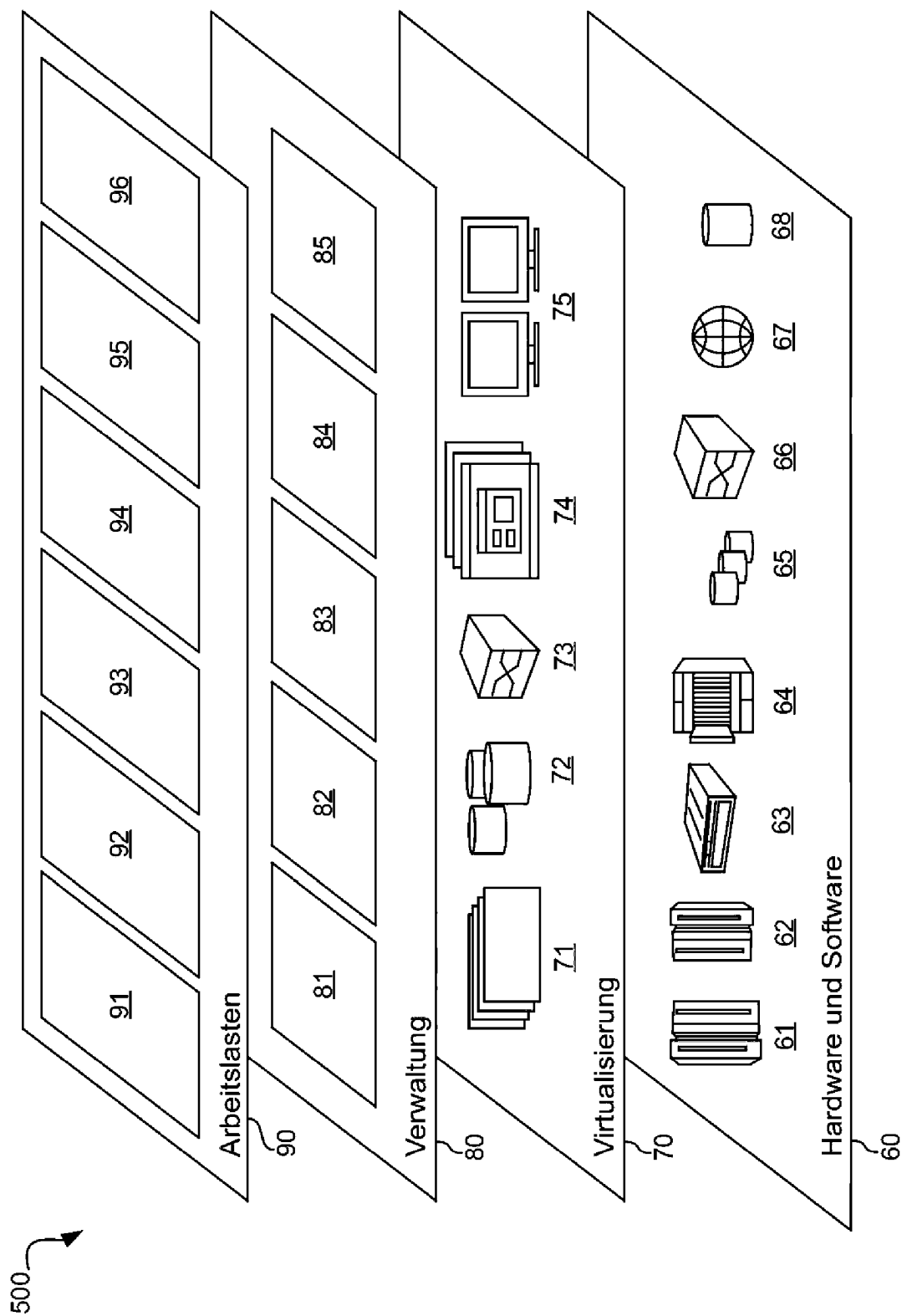


FIG. 5