



(12) **Veröffentlichung**

der internationalen Anmeldung mit der
(87) Veröffentlichungs-Nr.: **WO 2022/089199**
in der deutschen Übersetzung (Art. III § 8 Abs. 2
IntPatÜbkG)
(21) Deutsches Aktenzeichen: **11 2021 004 854.6**
(86) PCT-Aktenzeichen: **PCT/CN2021/123310**
(86) PCT-Anmeldetag: **12.10.2021**
(87) PCT-Veröffentlichungstag: **05.05.2022**
(43) Veröffentlichungstag der PCT Anmeldung
in deutscher Übersetzung: **07.09.2023**

(51) Int Cl.: **G06F 11/20 (2006.01)**
G06F 11/30 (2006.01)
G06F 11/34 (2006.01)

(30) Unionspriorität:
17/084,974 30.10.2020 US

(71) Anmelder:
**International Business Machines Corporation,
Armonk, NY, US**

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

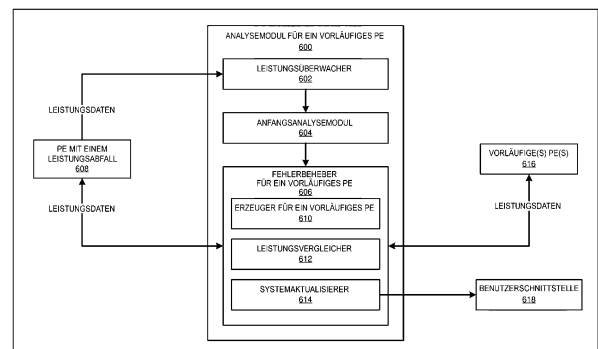
(72) Erfinder:
**Beuch, Daniel, Rochester, MN, US; Branson,
Michael J., Austin, TX, US; Stallman, Adam
Thomas, Rochester, MN, US; Cradick, Ryan K.,
Rochester, MN, US**

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **DYNAMISCHE ERSETZUNG VON VERARBEITUNGSELEMENTEN MIT EINEM
LEISTUNGSABFALL IN STREAMING-ANWENDUNGEN**

(57) Zusammenfassung: Bereitgestellt wird ein computerrealisiertes Verfahren. Das Verfahren enthält ein Überwachen einer verteilten Datenverarbeitungsanwendung während der Laufzeit auf ein Eintreten einer Bedingung, wobei die Bedingung ein Eintreten einer Leistungsabfallbedingung in einem Verarbeitungselement (Processing Element, PE) enthält. Das Verfahren enthält außerdem ein Starten eines vorläufigen PEs in einem zweiten Knoten während der Laufzeit, während eine Laufzeitoperation des PEs beibehalten wird. Das Verfahren vergleicht eine Operation des PEs mit einer Operation des vorläufigen PEs auf Grundlage einer Leistungsmetrik, die der Datenverarbeitungsressource zugehörig ist. Das Verfahren verbindet das vorläufige PE, um den Datenstrom von Tupeln von dem PE an ein nachgelagertes PE durch einen Datenstrom von Tupeln von dem vorläufigen PE zu dem nachgelagerten PE zu ersetzen.



Beschreibung

HINTERGRUND

[0001] Die vorliegende Erfindung bezieht sich im Allgemeinen auf ein Verfahren, System und Computerprogrammprodukt für ein Stream Computing. Im Besonderen bezieht sich die vorliegende Erfindung auf ein Verfahren, System und Computerprogrammprodukt für eine dynamische Ersetzung von Verarbeitungselementen in Streaming-Anwendungen.

[0002] Datenbanksysteme werden üblicherweise konfiguriert, um den Prozess eines Speicherns von Daten von einem Zugreifen, Verarbeiten oder Verwenden von Daten zu trennen, die in einer Datenbank gespeichert werden. Bei herkömmlichen, statischen Datenbanksystemen werden Daten vor einer anschließenden Abfrage und Analyse zunächst in einem Arbeitsspeicher gespeichert und indexiert. Im Allgemeinen sind solche statischen Datenbanksysteme nicht immer gut geeignet, um ein Verarbeiten und Analysieren von Streaming-Daten in Echtzeit durchzuführen. Zum Beispiel sind statische Datenbanksysteme mitunter nicht in der Lage, große Mengen von Streaming-Daten effizient oder in Echtzeit zu speichern, zu indexieren und zu analysieren.

[0003] Datenverarbeitung auf Datenstromgrundlage und Datenbankverarbeitung auf Datenstromgrundlage sind in den vergangenen Jahren als neue Technologien für Datenbanksysteme entstanden. Bei einer Datenstromanwendung ermöglichen Knoten, die über ein Netzwerk hinweg miteinander verbunden werden, Daten, von einem Knoten zum nächsten zu fließen. Solche Datenflüsse kapseln Blöcke von Daten in einem „Tupel“. Ein Tupel ist ein Block von Daten eines einzigen oder einer Vielfalt von verschiedenen Datentypen, z.B. Ganzzahl-, Gleitkomma-, boolesche oder Zeichenfolgedaten. Gruppen von Tupeln werden in Abfolgen übertragen, die als ein „Stream“ oder „Datenstrom“ bezeichnet werden. Insbesondere kann es vorkommen, dass Daten im Wesentlichen kontinuierlich als ein Datenstrom von Datenpunkten ankommen, die einem laufenden oder kontinuierlichen Ereignis entsprechen.

[0004] Zum Beispiel können Daten, die für den Börsenkurs einer bestimmten Aktie stehen, im Tagesverlauf im Allgemeinen schwanken, und ein Datenstromverwaltungssystem kann fortlaufend aktualisierte Börsenkurse empfangen, z.B. in gleichbleibenden Zeitabständen oder bei einer Änderung des Börsenkurses. Andere Beispiele für derartige Datenströme enthalten Temperatur- oder andere Umgebungsdaten, die von Sensoren erfasst werden, Computernetzwerk-Analysedaten, Patientendaten, die in einem Krankenhaus erfasst werden, oder auch Daten, die einen Herstellungsprozess oder

einen bzw. mehrere andere Geschäftsprozesse beschreiben.

KURZDARSTELLUNG

[0005] Die veranschaulichenden Ausführungsformen stellen eine dynamische Ersetzung von Verarbeitungselementen mit einem Leistungsabfall in Streaming-Anwendungen bereit. Eine Ausführungsform enthält ein Überwachen einer verteilten Datenverarbeitungsanwendung während der Laufzeit auf ein Eintreten einer Bedingung, wobei die verteilte Datenverarbeitungsanwendung einen ersten Knoten aufweist, der ein Verarbeitungselement (Processing Element, PE) enthält, das einen ersten Datenstrom von Tupeln von einem vorgelagerten PE empfängt, einen definierten Prozess für den ersten Datenstrom von Tupeln durchführt, der zu einem zweiten Datenstrom von Tupeln führt, und den zweiten Datenstrom von Tupeln an ein nachgelagertes PE überträgt, wobei die Bedingung ein Eintreten einer Leistungsabfallbedingung für das PE enthält. Die Ausführungsform enthält außerdem ein Starten eines ersten vorläufigen PEs in einem zweiten Knoten während der Laufzeit, während eine Laufzeitoperation des PEs beibehalten wird, wobei das erste vorläufige PE den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfängt und den definierten Prozess für den ersten Datenstrom von Tupeln durchführt. Die Ausführungsform enthält außerdem ein Vergleichen einer Operation des PEs mit einer Operation des ersten vorläufigen PEs auf Grundlage einer Leistungsmetrik, die der Datenverarbeitungsressource zugehörig ist. Die Ausführungsform enthält zudem ein Verbinden des ersten vorläufigen PEs, um den zweiten Datenstrom von Tupeln von dem PE an das nachgelagerte PE durch einen dritten Datenstrom von Tupeln von dem ersten vorläufigen PE zu dem nachgelagerten PE zu ersetzen. Andere Ausführungsformen dieses Aspekts enthalten entsprechende Computersysteme, Vorrichtungen und Computerprogramme, die in einer oder mehreren Computerspeichereinheiten aufgezeichnet werden, die jeweils konfiguriert werden, um die Aktionen der Ausführungsform durchzuführen.

[0006] Eine Ausführungsform enthält ein computernutzbare Programmprodukt. Das computernutzbare Programmprodukt enthält eine computerlesbare Speichereinheit und Programmanweisungen, die auf dem Speichermedium gespeichert werden.

[0007] Eine Ausführungsform enthält ein Computersystem. Das Computersystem enthält einen Prozessor, einen computerlesbaren Arbeitsspeicher und ein computerlesbares Speichermedium sowie auf dem Speichermedium gespeicherte Programmanweisungen zur Ausführung durch den Prozessor über den Arbeitsspeicher.

KURZBESCHREIBUNG DER ZEICHNUNGEN

[0008] Die neuartigen, für die Erfindung als kennzeichnend betrachteten Merkmale werden in den beigefügten Ansprüchen beschrieben. Die Erfindung selbst jedoch sowie eine bevorzugte Art der Verwendung, weitere Zielsetzungen und Vorzüge hiervon werden am deutlichsten unter Bezugnahme auf die folgende ausführliche Beschreibung der veranschaulichenden Ausführungsformen in Verbindung mit den beigefügten Zeichnungen, bei denen:

Fig. 1 ein Blockschaubild eines Netzwerks von Datenverarbeitungssystemen darstellt, in dem veranschaulichende Ausführungsformen realisiert werden können;

Fig. 2 ein Blockschaubild eines Datenverarbeitungssystems darstellt, in dem veranschaulichende Ausführungsformen realisiert werden können;

Fig. 3 ein Blockschaubild einer beispielhaften Datenstrom-Anwendungsumgebung gemäß einer veranschaulichenden Ausführungsform darstellt;

Fig. 4 ein Blockschaubild eines beispielhaften Datenverarbeitungsknotens gemäß einer veranschaulichenden Ausführungsform darstellt;

Fig. 5 ein Blockschaubild eines beispielhaften Verwaltungssystems gemäß einer veranschaulichenden Ausführungsform darstellt;

Fig. 6 ein Blockschaubild eines beispielhaften Analysemoduls für ein vorläufiges PE gemäß einer veranschaulichenden Ausführungsform darstellt;

Fig. 7 ein Blockschaubild eines Operatorgraphen für eine Stream-Computing-Anwendung gemäß einer veranschaulichenden Ausführungsform darstellt;

Fig. 8 ein Blockschaubild eines Operatorgraphen für eine Stream-Computing-Anwendung mit einem vorläufigen PE gemäß einer veranschaulichenden Ausführungsform darstellt;

Fig. 9 einen Ablaufplan eines beispielhaften dynamischen PE-Ersetzungsprozesses gemäß einer veranschaulichenden Ausführungsform darstellt;

Fig. 10 einen Ablaufplan eines beispielhaften Fehlerbehebungsprozesses für ein vorläufiges PE gemäß einer veranschaulichenden Ausführungsform darstellt;

Fig. 11 einen Ablaufplan eines beispielhaften Fehlerbehebungsprozesses für ein vorläufiges PE gemäß einer veranschaulichenden Ausführungsform darstellt;

Fig. 12 einen Ablaufplan eines beispielhaften Fehlerbehebungsprozesses für ein vorläufiges PE gemäß einer veranschaulichenden Ausführungsform darstellt.

AUSFÜHRLICHE BESCHREIBUNG

[0009] Datenverarbeitung auf Datenstromgrundlage und Datenbankverarbeitung auf Datenstromgrundlage sind in den vergangenen Jahren als eine neue Technologie für Datenbanksysteme entstanden. Es sind Produkte verfügbar, mit denen Benutzer Anwendungen erstellen können, die Streaming-Daten verarbeiten und abfragen, bevor sie eine Datenbankdatei erreichen. Mit dieser neu entstehenden Technologie können Benutzer eine Verarbeitungslogik festlegen, die auf eingehende Datensätze angewendet werden soll, während diese noch „auf dem Wege“ sind, sodass die Ergebnisse innerhalb einer sehr kurzen Zeitspanne verfügbar sind, oft innerhalb von Bruchteilen einer Sekunde. Ein Erstellen einer Anwendung unter Verwendung dieser Art von Verarbeitung hat ein neues Programmierparadigma eröffnet, das die Entwicklung einer breiten Vielfalt innovativer Anwendungen, Systeme und Prozesse ermöglicht, Anwendungsprogrammierer und Datenbankentwickler zugleich aber vor neue Herausforderungen stellt.

[0010] Die Hauptkomponenten von Datenstrom-Verarbeitungsanwendungen enthalten Tupel, Datenströme, Operatoren, PEs und Aufgaben. Ein „Tupel“ ist ein einzelnes Element der Daten in einem Datenstrom. Ein „Datenstrom“, wie hierin verwendet, bezieht sich auf eine fortlaufende Abfolge von Tupeln. Ein „Operator“, wie hierin verwendet, ist eine logische Funktion, die die Tupeldaten aus dem eingehenden Datenstrom verarbeitet und die Ergebnisse in Form eines Ausgabedatenstroms erzeugt. Ein „Operatorgraph“, wie hierin verwendet, ist eine visuelle Darstellung der Operatoren, die durch die Datenströme verbunden werden, die durch sie hindurchfließen, und definiert die Analyseanwendung. Ein „Verarbeitungselement“ oder „PE“, wie hierin verwendet, ist ein ausführbares Element, das einen Satz von einem oder mehreren Operatoren enthält, die in demselben Prozess in einem Netzwerkknoten oder einer Datenverarbeitungsressource ausgeführt werden, z.B. einem Server, Client, Host, Container oder einer anderen Datenverarbeitungseinheit wie z.B. einem Datenverarbeitungssystem. Ein „Knoten“, wie hierin verwendet, ist ein Netzwerkknoten wie zum Beispiel eine beliebige elektronische Einheit, die über ein Computernetzwerk adressierbar ist, der in der Lage ist, Informationen über das Computernetzwerk zu erzeugen, zu empfangen und/oder zu übertragen, und kann sich auf ein Element, ein Modul, eine Komponente, eine Platine, eine Einheit oder ein System beziehen.

[0011] Eine Datenstrom-Verarbeitungsaufgabe hat einen gerichteten Graphen von PEs, die Datentupel zwischen den PEs senden. Das PE verarbeitet die eingehenden Tupel und erzeugt Ausgabentupel. Ein PE hat eine unabhängige Verarbeitungseinheit und wird auf einem Host ausgeführt. Die Datenstromplattform kann aus einer Zusammenstellung von Hosts bestehen, die für die Aufnahme von PEs in Frage kommen.

[0012] Bei einer Stream-Computing-Anwendung werden PEs daher so miteinander verbunden, dass Daten über ein Netzwerk (z.B. über ein TCP/IP-Socket) von einem PE zum nächsten fließen. Daten fließen in Form eines Tupels von einem zu einem weiteren Datenstromoperator, wobei das Tupel eine Abfolge von einem oder mehreren Attributen hat, die einer Entität zugehörig sind.

[0013] Stream-Computing-Anwendungen bewältigen sehr große Datenvolumen, die effizient und in Echtzeit verarbeitet werden müssen. Stream Computing ist in der Lage, ein Streaming und eine Skalierbarkeit mit einer hohen Leistung zu erreichen, indem eine Anwendung auf mehrere Knoten verteilt wird, indem ausführbare Elemente (d.h. PEs) erzeugt werden, sowie PEs in mehreren Knoten repliziert werden und ein Lastausgleich zwischen ihnen durchgeführt wird. Auf diese Weise kann eine Stream-Computing-Anwendung kontinuierlich Hunderttausende von Nachrichten pro Sekunde und bis zu mehrere Petabytes an Daten pro Tag aufnehmen und analysieren. Entsprechend muss jeder Datenstromoperator in einer Stream-Computing-Anwendung ein empfangenes Tupel unter Umständen innerhalb von Bruchteilen einer Sekunde verarbeiten können.

[0014] Ein Vorteil von Produkten wie IBM Streams ist die enorme Skalierbarkeit und Leistung (IBM ist eine eingetragene Marke der International Business Machines Corporation). Leistungsvorteile enthalten Verbesserungen hinsichtlich der Fähigkeit, potenziell noch sehr viel größere Mengen an kontinuierlichen Datenströmen innerhalb einer bestimmten Zeitspanne aufzunehmen, zu filtern, zu analysieren und zu korrelieren, als dies bisher möglich war. Wenn eine Streaming-Anwendung bei ihrer Verarbeitung jedoch „in Verzug gerät“, büßt die von ihr durchgeführte Analyse ihren Wert ein, d.h. Ergebnisse der Analyse beruhen auf den vergangenen, nicht auf den neuesten Daten.

[0015] Eine Streaming-Anwendung kann einfach deswegen in Verzug geraten, weil ein Teil der Anwendung eine mangelnde Leistung aufweist. Eine Streaming-Anwendung kann auch aufgrund von Fehlern innerhalb der Anwendung in Verzug geraten. Ein Anbieter von Streaming-Diensten oder ein Betriebssystem für Streaming-Dienste kann zwar eine Hochverfügbarkeit und eine Anwendungs-

funktionsübernahme unterstützen, Fehlerbehebungsmechanismus enthalten jedoch eine Zeitspanne, in der Daten nicht verarbeitet werden, während die Fehlerbehebung stattfindet. So kann die Zeitspanne, die für einen Ausfall und Neustart eines Verarbeitungselements (PEs) notwendig ist, von 20 bis 30 Sekunden bis hin zu mehreren Minuten dauern, abhängig von einer Anzahl von Faktoren (z.B. ein schneller Ausfall im Gegensatz zu einem schrittweisen Versagen, bei dem das PE immer langsamer wird, bis es schließlich ausfällt, die Zeitspanne zum Erzeugen/Einplanen einer neuen Ressource, um ein neu gestartetes PE zu hosten, die Zeitspanne zum Bereitstellen des neuen PEs usw.).

[0016] Je länger der Fehlerbehebungsprozess dauert, desto größer ist der Rückstau an nicht verarbeiteten Echtzeitdaten, wodurch die Anwendung hinter die Echtzeit zurückfällt, bis sie wieder zu den neuesten Daten „aufschließt“. Wenn mehrere Fehler auftreten, kann es zu lang andauernden Fehlerbehebungszeiträumen für die Anwendung kommen, sodass diese vollständig durch Echtzeitdaten überschwemmt wird und dadurch komplett ausfällt.

[0017] Die veranschaulichenden Ausführungsformen der vorliegenden Offenbarung betreffen Probleme und Leistungsziele, die spezifisch für eine Datenstromverarbeitung sind, indem deren Verarbeitungselemente (PEs) auf einen Leistungsabfall überwacht und ein PE mit einem Leistungsabfall (vor seinem Ausfall) durch eine besser funktionierende Kopie seiner selbst ersetzt wird. Einige derartige Ausführungsformen stellen einen oder mehrere Vorteile gegenüber früheren Methoden bereit, z.B. ein Verbessern der Leistung einer Streaming-Anwendung, indem PEs mit einer unterdurchschnittlichen Leistung ersetzt werden, ein Bereitstellen einer Art und Weise, wie sich PE-Fehler und zugehörige, lang andauernde Fehlerbehebungen vermeiden lassen, indem ein PE mit einem Leistungsabfall vor seinem Ausfall durch eine besser funktionierende Kopie seiner selbst ersetzt wird, und ein Ermöglichen eines Umschaltens auf ein Ersatz-PE, das sehr viel schneller als ein vollständiges Fehlerbehebungsszenario ist, wodurch der Aufstau von nicht verarbeiteten Echtzeitdaten während des Fehlerbehebungsprozesses verringert oder verhindert wird.

[0018] Bei einer veranschaulichenden Ausführungsform überwacht ein Datenstromverwalter eine Streaming-Anwendung, indem er Statistiken verwaltet, die für die Leistung der verschiedenen PEs stehen, aus denen die Streaming-Anwendung besteht. Bei manchen Ausführungsformen überwacht der Datenstromverwalter PEs auf einen Leistungsabfall und führt, wenn ein Leistungsabfall eines oder mehrerer PEs erkannt wird, Fehlerbehebungsprozesse durch oder gibt Benachrichtigungssignale oder -daten an ein Fehlerbehebungsmodul oder eine separate

Anwendung aus. Zum Beispiel überwacht bei manchen Ausführungsformen ein PE sich selbst auf einen Leistungsabfall und sendet ein Signal an einen Datenstromverwalter, wenn das PE einen Leistungsabfall erkennt.

[0019] Bei manchen Ausführungsformen erkennt ein Datenstromverwalter einen Leistungsabfall eines PEs, indem er das PE auf ein Eintreten einer Bedingung überwacht, die als ein Leistungsabfallindikator dient, der sich von einem PE zu einem weiteren PE unterscheiden kann. Bei manchen Ausführungsformen können sich auch die Schwellenwerte für Leistungsabfallindikator-Bedingungen von einem PE zu einem weiteren PE unterscheiden. Bei manchen derartigen Ausführungsformen ist eine Leistungsabfallindikator-Bedingung somit für jedes PE konfigurierbar.

[0020] Bei manchen Ausführungsformen enthält eine Leistungsabfallindikator-Bedingung Ausführungsbedingungen, die einem PE zugehörig sind und von einem vordefinierten oder erwarteten Muster abweichen. Zum Beispiel enthalten bei manchen Ausführungsformen Indikatoren für einen Leistungsabfall eine anomale Anzahl von Tupeln in Warteschlangen (d.h. Tupeln, die an Eingabeanschlüssen von Operatoren in dem PE darauf warten, verarbeitet zu werden), eine anomale Arbeitsspeichernutzung, Anzahl und Arten von verarbeiteten Ausnahmen sowie eine anomale CPU-Auslastung.

[0021] Bei manchen Ausführungsformen überwacht der Datenstromverwalter eine Mehrzahl von PEs. Bei alternativen Ausführungsformen überwacht der Datenstromverwalter nur ein einziges PE. Bei manchen Ausführungsformen verwaltet der Datenstromverwalter Statistiken, die für die normale oder erwartete Leistung des PEs vor einem Leistungsabfall des PEs stehen. Bei manchen Ausführungsformen protokolliert der Datenstromverwalter die Nutzung derartiger Datenverarbeitungsressourcen durch das PE, wodurch normale Nutzungsmuster erstellt werden können und wodurch zudem anomale Nutzungsmuster erkannt werden können.

[0022] Bei manchen Ausführungsformen enthält eine anomale Anzahl von Tupeln in Warteschlangen eine Anzahl von Tupeln in Warteschlangen, die höher als normal ist. Bei manchen Ausführungsformen enthält eine anomale Arbeitsspeichernutzung anomal hohe Änderungsraten bei der Arbeitsspeichernutzung. Bei manchen Ausführungsformen enthält eine anomale CPU-Auslastung anomal hohe Änderungsraten bei der CPU-Auslastung.

[0023] Wenn ein Datenstromverwalter einen Leistungsabfall eines PEs erkennt, führt der Datenstromverwalter bei manchen Ausführungsformen eine Anfangsanalyse in Verbindung mit dem PE mit

einem Leistungsabfall durch. Bei manchen derartigen Ausführungsformen analysiert der Datenstromverwalter das PE mit einem Leistungsabfall, um zu ermitteln, ob die Anzeichen für die nachlassende Leistung tatsächlich Anzeichen für ein anderes Problem als ein Leistungsabfall des PEs mit einem Leistungsabfall sind, z.B. für einen Rückstau oder eine oder mehrere Cluster-Datenverarbeitungsressourcen, die ihre Kapazitätsgrenzen erreichen.

[0024] Wenn ein Datenstromverwalter einen Leistungsabfall eines PEs erkennt und kein alternatives Problem erkennt, kann der Datenstromverwalter bei manchen derartigen Ausführungsformen versuchen, ein oder mehrere vorläufige PEs als möglichen Ersatz für das PE mit einem Leistungsabfall zu verwenden. Bei manchen derartigen Ausführungsformen tauscht der Datenstromverwalter mit einem Einplaner Daten aus, um einen Knoten zum Hosten des vorläufigen PEs zu identifizieren. Bei manchen Ausführungsformen sucht der Einplaner nach einem anderen Knoten als dem Knoten, der das PE hostet. Bei manchen Ausführungsformen sucht der Einplaner auch auf Grundlage von vorgegebenen Regeln oder Kriterien nach einem Knoten, zum Beispiel einer Verfügbarkeit von Datenverarbeitungsressourcen und einer Nähe zu vor- und nachgelagerten PEs. Sobald der Einplaner einen Host-Knoten identifiziert, benachrichtigt der Einplaner bei manchen Ausführungsformen den Datenstromverwalter von dem ausgewählten Speicherort für das vorläufige PE.

[0025] Bei manchen Ausführungsformen startet der Datenstromverwalter das vorläufige PE in dem identifizierten Knoten während der Laufzeit, während eine Laufzeitoperation des PEs mit einem Leistungsabfall beibehalten wird. Bei manchen derartigen Ausführungsformen verbindet der Datenstromverwalter das vorläufige PE mit einem vorgelagerten PE, das dasselbe vorgelagerte PE von dem PE mit einem Leistungsabfall ist, sodass sowohl das vorläufige PE als auch das PE mit einem Leistungsabfall zur selben Zeit denselben Datenstrom von Tupeln von demselben vorgelagerten PE empfangen. Bei manchen derartigen Ausführungsformen führen sowohl das vorläufige PE als auch das PE mit einem Leistungsabfall denselben definierten Prozess für den Datenstrom von Tupeln durch, wodurch der Datenstromverwalter die Leistung des PEs mit einem Leistungsabfall mit derjenigen des vorläufigen PEs vergleichen kann. Bei manchen derartigen Ausführungsformen führt das PE mit einem Leistungsabfall den definierten Prozess für den Datenstrom von Tupeln durch, was zu einem Ausgabedatenstrom von Tupeln führt, den das PE mit einem Leistungsabfall an ein nachgelagertes PE überträgt, während die Ausgabe des vorläufigen PEs mit keinerlei nachgelagerten Elementen verbunden wird.

[0026] Bei manchen Ausführungsformen vergleicht der Datenstromverwalter die Operation des PEs mit einem Leistungsabfall mit derjenigen des vorläufigen PEs auf Grundlage einer Leistungsmetrik, die einer oder mehreren Datenverarbeitungsressourcen zugehörig ist. Zum Beispiel vergleicht der Datenstromverwalter bei manchen Ausführungsformen die Operation des PEs mit einem Leistungsabfall mit derjenigen des vorläufigen PEs, indem er eines oder mehrere einer CPU-Auslastung, einer Arbeitsspeichernutzung und einer Geschwindigkeit des Tupeldurchsatzes vergleicht. Wenn der Datenstromverwalter ermittelt, dass das vorläufige PE eine bessere Leistung als das PE mit einem Leistungsabfall aufweist, ersetzt der Datenstromverwalter bei manchen derartigen Ausführungsformen das PE mit einem Leistungsabfall durch das vorläufige PE, zum Beispiel, indem die Ausgabe des vorläufigen PEs anstatt des PEs mit einem Leistungsabfall mit dem nachgelagerten PE verbunden wird, und der Datenstromverwalter beendet das PE mit einem Leistungsabfall. Bei manchen Ausführungsformen sendet der Datenstromverwalter über eine Benutzerschnittstelle eine Benachrichtigung über den Ersatz des PEs mit einem Leistungsabfall durch das vorläufige PE an den Benutzer.

[0027] Aus Gründen der Klarheit der Beschreibung und ohne eine wie auch immer geartete Beschränkung darauf nahezulegen, werden die veranschaulichenden Ausführungsformen unter Verwendung einiger Beispielkonfigurationen beschrieben. Ausgehend von dieser Offenbarung können Fachleute viele Abwandlungen, Anpassungen und Modifikationen an einer beschriebenen Konfiguration vornehmen, um einen beschriebenen Zweck zu erreichen, und diese werden als innerhalb des inhaltlichen Umfangs der veranschaulichenden Ausführungsformen liegend betrachtet.

[0028] Darüber hinaus werden in den Figuren und den veranschaulichenden Ausführungsformen vereinfachte Darstellungen der Datenverarbeitungsumgebungen verwendet. In einer tatsächlichen Datenverarbeitungsumgebung können zusätzliche Strukturen oder Komponenten, die hierin nicht gezeigt oder beschrieben werden, oder Strukturen oder Komponenten vorhanden sein, die sich von den hierin beschriebenen unterscheiden, jedoch einer ähnlichen Funktion dienen, ohne vom inhaltlichen Umfang der veranschaulichenden Ausführungsformen abzuweichen.

[0029] Darüber hinaus werden die veranschaulichenden Ausführungsformen lediglich beispielhaft in Bezug auf spezifische tatsächliche oder hypothetische Komponenten beschrieben. Die durch die verschiedenen veranschaulichenden Ausführungsformen beschriebenen Schritte können angepasst werden, um Erläuterungen zu Entscheidungen

bereitzustellen, die zum Beispiel durch ein Klassifikatormodell für maschinelles Lernen getroffen werden.

[0030] Alle spezifischen Ausprägungen dieser und anderer ähnlicher Gegenstände sind nicht als Beschränkung der Erfindung zu verstehen. Innerhalb des inhaltlichen Umfangs der veranschaulichenden Ausführungsformen kann jegliche geeignete Ausprägung dieser und anderer ähnlicher Gegenstände gewählt werden.

[0031] Die Beispiele in dieser Offenbarung dienen lediglich der Klarheit der Beschreibung und stellen keine Beschränkung der veranschaulichenden Ausführungsformen dar. Etwaige hierin genannte Vorzüge dienen lediglich als Beispiele und sind nicht als Beschränkung der veranschaulichenden Ausführungsformen gedacht. Durch spezifische veranschaulichende Ausführungsformen können zusätzliche oder abweichende Vorzüge realisiert werden. Darüber hinaus kann eine bestimmte veranschaulichende Ausführungsform einige, alle oder keinen der oben genannten Vorzüge haben.

[0032] Darüber hinaus können die veranschaulichenden Ausführungsformen in Bezug auf jegliche Art von Daten, Datenquelle oder Zugriff auf eine Datenquelle über ein Datennetzwerk realisiert werden. Innerhalb des inhaltlichen Umfangs der Erfindung kann jegliche Art von Datenspeichereinheit die Daten einer Ausführungsform der Erfindung bereitstellen, entweder lokal in einem Datenverarbeitungssystem oder über ein Datennetzwerk. Wo eine Ausführungsform unter Verwendung einer mobilen Einheit beschrieben wird, kann innerhalb des inhaltlichen Umfangs der veranschaulichenden Ausführungsformen jegliche Art von Datenspeichereinheit, die für eine Verwendung mit der mobilen Einheit geeignet ist, die Daten für eine solche Ausführungsform bereitstellen, entweder lokal in der mobilen Einheit oder über ein Datennetzwerk.

[0033] Die veranschaulichenden Ausführungsformen werden lediglich als Beispiele unter Verwendung von spezifischem Code, kontrastiven Erläuterungen, einem computerlesbaren Speichermedium, allgemeinen Merkmalen, historischen Daten, Entwürfen, Architekturen, Protokollen, Gestaltungen, Schemata und Werkzeugen beschrieben und stellen keine Beschränkung auf die veranschaulichenden Ausführungsformen dar. Darüber hinaus werden die veranschaulichenden Ausführungsformen in manchen Fällen unter Verwendung von bestimmter Software, bestimmten Werkzeugen und Datenverarbeitungsumgebungen beschrieben, wobei dies lediglich als ein Beispiel zur besseren Klarheit der Beschreibung dient. Die veranschaulichenden Ausführungsformen können gemeinsam mit anderen vergleichbaren oder für einen ähnlichen Zweck gedachten Strukturen, Systemen, Anwendungen

oder Architekturen verwendet werden. Zum Beispiel können innerhalb des inhaltlichen Umfangs der Erfindung andere vergleichbare mobile Einheiten, Strukturen, Systeme, Anwendungen oder Architekturen gemeinsam mit einer solchen Ausführungsform der Erfindung verwendet werden. Eine veranschaulichende Ausführungsform kann in Hardware, Software oder einer Kombination hiervon realisiert werden.

[0034] Die Beispiele in dieser Offenbarung dienen lediglich der Klarheit der Beschreibung und stellen keine Beschränkung der veranschaulichenden Ausführungsformen dar. Ausgehend von dieser Offenbarung sind zusätzliche Daten, Operationen, Aktionen, Aufgaben, Aktivitäten und Bearbeitungen denkbar, die als innerhalb des inhaltlichen Umfangs der veranschaulichenden Ausführungsformen liegend betrachtet werden.

[0035] Etwaige hierin genannte Vorzüge dienen lediglich als Beispiele und sind nicht als Beschränkung der veranschaulichenden Ausführungsformen gedacht. Durch spezifische veranschaulichende Ausführungsformen können zusätzliche oder abweichende Vorzüge realisiert werden. Darüber hinaus kann eine bestimmte veranschaulichende Ausführungsform einige, alle oder keinen der oben genannten Vorzüge aufweisen.

[0036] Unter Bezugnahme auf die Figuren und im Besonderen auf die **Fig. 1** und **Fig. 2** handelt es sich bei diesen Figuren um Beispieldarstellungen von Datenverarbeitungsumgebungen, in denen veranschaulichende Ausführungsformen realisiert werden können. Die **Fig. 1** und **Fig. 2** stellen lediglich Beispiele dar und sollen mit Blick auf die Umgebungen, in denen verschiedene Ausführungsformen realisiert werden können, keinerlei Beschränkung geltend machen oder nahelegen. Eine bestimmte Realisierung kann auf Grundlage der folgenden Beschreibung viele Modifikationen an den dargestellten Umgebungen vornehmen.

[0037] **Fig. 1** stellt ein Blockschaubild eines Netzwerks von Datenverarbeitungssystemen dar, in dem veranschaulichende Ausführungsformen realisiert werden können. Eine Datenverarbeitungsumgebung 100 ist ein Netzwerk von Computern, in dem die veranschaulichenden Ausführungsformen realisiert werden können. Die Datenverarbeitungsumgebung 100 enthält ein Netzwerk 102. Das Netzwerk 102 ist das Medium, mit dem Datenübertragungsverbindungen zwischen verschiedenen Einheiten und Computern bereitgestellt werden, die innerhalb der Datenverarbeitungsumgebung 100 miteinander verbunden werden. Das Netzwerk 102 kann Verbindungen wie z.B. drahtgebundene und drahtlose Datenübertragungsverbindungen oder Lichtwellenleiterkabel enthalten.

[0038] Clients und Server sind lediglich Beispielfunktionen bestimmter Datenverarbeitungssysteme, die mit dem Netzwerk 102 verbunden werden, und sollen andere Konfigurationen oder Funktionen dieser Datenverarbeitungssysteme nicht ausschließen. Ein Datenverarbeitungssystem 104 wird mit dem Netzwerk 102 verbunden. Software-Anwendungen können in einem beliebigen Datenverarbeitungssystem in der Datenverarbeitungsumgebung 100 ausgeführt werden. Jede Software-Anwendung, die als in dem Datenverarbeitungssystem 104 aus **Fig. 1** ausgeführt beschrieben wird, kann konfiguriert werden, um auf ähnliche Weise in einem weiteren Datenverarbeitungssystem ausgeführt zu werden. Jegliche Daten oder Informationen, die in dem Datenverarbeitungssystem 104 aus **Fig. 1** gespeichert oder erzeugt werden, können konfiguriert werden, um auf ähnliche Weise in einem weiteren Datenverarbeitungssystem ausgeführt oder erzeugt zu werden. Ein Datenverarbeitungssystem wie z.B. das Datenverarbeitungssystem 104 kann Daten enthalten und Software-Anwendungen oder Software-Werkzeuge haben, die darauf Datenverarbeitungsprozesse ausführen. Bei einer Ausführungsform enthält das Datenverarbeitungssystem 104 einen Arbeitsspeicher 124, der eine Anwendung 105A enthält, die konfiguriert werden kann, um eine oder mehrere der Datenprozessorfunktionen zu realisieren, die hierin gemäß einer oder mehreren Ausführungsformen beschrieben werden.

[0039] Ein Server 106 wird zusammen mit einer Speichereinheit 108 mit dem Netzwerk 102 verbunden. Die Speichereinheit 108 enthält eine Datenbank 109, die konfiguriert wird, um Daten zu speichern, wie hierin in Bezug auf verschiedene Ausführungsformen beschrieben wird, zum Beispiel Bilddaten und Attributdaten. Der Server 106 ist ein herkömmliches Datenverarbeitungssystem. Bei einer Ausführungsform enthält der Server 106 PEs einer Datenstrom-Verarbeitungsanwendung 105B, die konfiguriert werden können, um eine oder mehrere der Prozessorfunktionen zu realisieren, die hierin gemäß einer oder mehreren Ausführungsformen beschrieben werden.

[0040] Auch Clients 110, 112 und 114 werden mit dem Netzwerk 102 verbunden. Ein herkömmliches Datenverarbeitungssystem wie z.B. der Server 106 oder der Client 110, 112 oder 114 kann Daten enthalten und Software-Anwendungen oder Software-Werkzeuge haben, die darauf herkömmliche Datenverarbeitungsprozesse ausführen.

[0041] Lediglich beispielhaft und ohne eine wie auch immer geartete Beschränkung auf eine solche Architektur nahezu legen, stellt **Fig. 1** bestimmte Komponenten dar, die in einer Beispielrealisierung einer Ausführungsform verwendbar sind. Zum Beispiel werden der Server 106 und die Clients 110, 112,

114 nur beispielhaft als Server und Clients dargestellt und sollen keine Beschränkung auf eine Client-Server-Architektur bedeuten. Als ein weiteres Beispiel kann eine Ausführungsform, wie gezeigt, auf mehrere Datenverarbeitungssysteme und ein Datennetzwerk verteilt werden, wohingegen eine weitere Ausführungsform innerhalb des inhaltlichen Umfangs der veranschaulichenden Ausführungsformen in einem einzigen Datenverarbeitungssystem realisiert werden kann. Die herkömmlichen Datenverarbeitungssysteme 106, 110, 112 und 114 stehen außerdem für Beispielknoten in einem Cluster, für Partitionen und andere Konfigurationen, die zum Realisieren einer Ausführungsform geeignet sind.

[0042] Eine Einheit 132 ist ein Beispiel für eine hierin beschriebene herkömmliche Datenverarbeitungseinheit. Zum Beispiel kann die Einheit 132 in Gestalt eines Smartphones, eines Tablet Computers, eines Laptop Computers, eines Clients 110 in einer ortsgebundenen oder einer tragbaren Form, einer am Körper tragbaren Datenverarbeitungseinheit oder jeder anderen geeigneten Einheit vorliegen. Bei einer Ausführungsform sendet die Einheit 132 Anforderungen an den Server 106, um eine oder mehrere Datenverarbeitungsaufgaben durch die Datenstrom-Verarbeitungsanwendung 105B durchzuführen, z.B. ein Initiieren von hierin beschriebenen Prozessen. Jede Software-Anwendung, die als in einem weiteren herkömmlichen Datenverarbeitungssystem aus **Fig. 1** ausgeführt beschrieben wird, kann konfiguriert werden, um auf ähnliche Weise in der Einheit 132 ausgeführt zu werden. Jegliche Daten oder Informationen, die in einem weiteren herkömmlichen Datenverarbeitungssystem aus **Fig. 1** gespeichert oder ausgeführt werden, können konfiguriert werden, um auf ähnliche Weise in der Einheit 132 ausgeführt zu werden.

[0043] Der Server 106, die Speichereinheit 108, das Datenverarbeitungssystem 104 und die Clients 110, 112 und 114 sowie die Einheit 132 können unter Verwendung von drahtgebundenen Verbindungen, Protokollen für eine drahtlose Datenübertragung oder einer anderen geeigneten Datenverbindung mit dem Netzwerk 102 verbunden werden. Die Clients 110, 112 und 114 können zum Beispiel Personal Computer oder Netzwerkcomputer sein.

[0044] In dem abgebildeten Beispiel kann der Server 106 den Clients 110, 112 und 114 Daten wie z.B. Boot-Dateien, Betriebssystemabbilder und Anwendungen bereitstellen. In diesem Beispiel können die Clients 110, 112 und 114 Clients des Servers 106 sein. Die Clients 110, 112, 114 oder eine Kombination hiervon können ihre eigenen Daten, Boot-Dateien, Betriebssystemabbilder und Anwendungen enthalten. Die Datenverarbeitungsumgebung 100 kann zusätzliche Server, Clients und andere nicht gezeigte Einheiten enthalten.

[0045] In dem dargestellten Beispiel kann der Arbeitsspeicher 124 dem Prozessor 122 Daten wie z.B. Boot-Dateien, Betriebssystemabbilder und Anwendungen bereitstellen. Der Prozessor 122 kann seine eigenen Daten, Boot-Dateien, Betriebssystemabbilder und Anwendungen enthalten. Die Datenverarbeitungsumgebung 100 kann zusätzliche Arbeitsspeicher, Prozessoren und andere nicht gezeigte Einheiten enthalten.

[0046] In dem dargestellten Beispiel kann die Datenverarbeitungsumgebung 100 das Internet sein. Das Netzwerk 102 kann für eine Zusammenstellung von Netzwerken und Gateways stehen, die das TCP/IP-Protokoll (Transmission Control Protocol/Internet Protocol) und andere Protokolle verwenden, um untereinander Daten auszutauschen. Der Kern des Internets wird aus einer Struktur von Datenübertragungsverbindungen zwischen wichtigen Knoten oder Host-Computern gebildet, die Tausende von gewerblichen, behördlichen, dem Bildungssektor zugehörigen und anderen Computersystemen enthalten, die Daten und Nachrichten weiterleiten. Selbstverständlich kann die Datenverarbeitungsumgebung 100 auch als eine Anzahl verschiedener Arten von Netzwerken realisiert werden, zum Beispiel als ein Intranet, ein lokales Netzwerk (Local Area Network, LAN) oder ein Weitverkehrsnetzwerk (Wide Area Network, WAN). **Fig. 1** ist als Beispiel und nicht als architektonische Beschränkung der verschiedenen veranschaulichenden Ausführungsformen gedacht.

[0047] Neben anderen Verwendungen kann die Datenverarbeitungsumgebung 100 zum Realisieren einer Client-Server-Umgebung verwendet werden, in der die veranschaulichenden Ausführungsformen realisiert werden können. Eine Client-Server-Umgebung ermöglicht eine Verteilung von Software-Anwendungen und Daten in einem Netzwerk, sodass eine Anwendung funktioniert, indem sie die Wechselwirkung zwischen einem herkömmlichen Client-Datenverarbeitungssystem und einem herkömmlichen Server-Datenverarbeitungssystem nutzt. Die Datenverarbeitungsumgebung 100 kann außerdem eine dienstorientierte Architektur nutzen, bei der in einem Netzwerk verteilte interoperable Software-Komponenten als zusammenhängende Geschäftsanwendungen gebündelt werden können. Die Datenverarbeitungsumgebung 100 kann auch in Gestalt einer Cloud vorliegen und ein Cloud-Computing-Modell einer Dienstbereitstellung nutzen, um einen problemlosen, bedarfsgesteuerten Netzwerkzugriff auf einen gemeinsam genutzten Vorrat von konfigurierbaren Datenverarbeitungsressourcen (z.B. Netzwerke, Netzwerkbandbreite, Server, Verarbeitung, Arbeitsspeicher, Speicher, Anwendungen, virtuelle Maschinen und Dienste) zu ermöglichen, die mit minimalem Verwaltungsaufwand bzw. minimaler

Interaktion mit einem Anbieter des Dienstes schnell bereitgestellt und freigegeben werden können.

[0048] Unter Bezugnahme auf **Fig. 2** stellt diese Figur ein Blockschaubild eines Datenverarbeitungssystems dar, in dem veranschaulichende Ausführungsformen realisiert werden können. Ein Datenverarbeitungssystem 200 ist ein Beispiel für einen herkömmlichen Computer wie z.B. das Datenverarbeitungssystem 104, den Server 106 oder die Clients 110, 112 und 114 aus **Fig. 1** oder eine weitere Art von Einheit, in der sich computernutzbarer Programmcode bzw. computernutzbare Anweisungen, die die Prozesse realisieren, für die veranschaulichenden Ausführungsformen befinden können.

[0049] Das Datenverarbeitungssystem 200 ist außerdem repräsentativ für ein herkömmliches Datenverarbeitungssystem oder eine darin befindliche Konfiguration wie z.B. das herkömmliche Datenverarbeitungssystem 132 aus **Fig. 1**, in dem sich computernutzbarer Programmcode oder computernutzbare Anweisungen befinden können, welche die Prozesse der veranschaulichenden Ausführungsformen realisieren. Das Datenverarbeitungssystem 200 wird nur beispielhaft als ein Computer beschrieben, ohne hierauf beschränkt zu sein. Realisierungen in Gestalt anderer Einheiten wie z.B. der Einheit 132 aus **Fig. 1** können das Datenverarbeitungssystem 200 abwandeln, indem sie beispielsweise eine berührungsempfindliche Schnittstelle hinzufügen, und sogar bestimmte dargestellte Komponenten des Datenverarbeitungssystems 200 weglassen, ohne von der allgemeinen Beschreibung der Operationen und Funktionen des hierin beschriebenen Datenverarbeitungssystems 200 abzuweichen.

[0050] In dem dargestellten Beispiel verwendet das Datenverarbeitungssystem 200 eine Hub-Architektur, die einen North Bridge & Memory Controller Hub (NB/MCH) 202 und einen South Bridge & Input/Output (I/O) Controller Hub (SB/ICH) 204 enthält. Eine Verarbeitungseinheit 206, ein Hauptspeicher 208 und ein Grafikprozessor 210 werden mit dem North Bridge & Memory Controller Hub (NB/MCH) 202 verbunden. Die Verarbeitungseinheit 206 kann einen oder mehrere Prozessoren enthalten und unter Verwendung eines oder mehrerer heterogener Prozessorsysteme realisiert werden. Die Verarbeitungseinheit 206 kann ein Mehrkernprozessor sein. Bei bestimmten Ausführungsformen kann der Grafikprozessor 210 über einen Accelerated Graphics Port (AGP) mit dem NB/MCH 202 verbunden werden.

[0051] In dem dargestellten Beispiel wird ein LAN-Adapter 212 mit dem South Bridge & I/O Controller Hub (SB/ICH) 204 verbunden. Ein Audioadapter 216, ein Tastatur- und Mausadapter 220, ein Modem 222,

ein Festwertspeicher (Read Only Memory, ROM) 224, USB-Anschlüsse (Universal Serial Bus) und andere Anschlüsse 232 sowie PCI/PCIe-Einheiten 234 werden über den Bus 238 mit dem South Bridge & I/O Controller Hub 204 verbunden. Ein Festplattenlaufwerk (Hard Disk Drive, HDD) oder eine Halbleiterplatte (Solid State Drive, SSD) 226 und ein CD-ROM 230 werden über einen Bus 240 mit dem South Bridge & I/O Controller Hub 204 verbunden. Die PCI/PCIe-Einheiten 234 können zum Beispiel Ethernet-Adapter, Erweiterungskarten und PC-Karten für Notebook Computer enthalten. PCI verwendet einen CardBus-Controller, PCIe nicht. Der ROM 224 kann zum Beispiel ein Flash-BIOS (Binary Input/Output System) sein. Das Festplattenlaufwerk 226 und der CD-ROM 230 können zum Beispiel eine IDE-Schnittstelle (Integrated Drive Electronics), eine SATA-Schnittstelle (Serial Advanced Technology Attachment) oder Abwandlungen wie z.B. external SATA (eSATA) und Micro-SATA (mSATA) verwenden. Eine SIO-Einheit (Super I/O) 236 kann über einen Bus 238 mit dem South Bridge & I/O Controller Hub (SB/ICH) 204 verbunden werden.

[0052] Arbeitsspeicher wie z.B. der Hauptspeicher 208, der ROM 224 oder ein (nicht gezeigter) Flash-Arbeitsspeicher sind einige Beispiele für computernutzbare Speichereinheiten. Das Festplattenlaufwerk oder die Halbleiterplatte 226, der CD-ROM 230 und andere ähnlich nutzbare Einheiten sind einige Beispiele für computernutzbare Speichereinheiten mit einem computernutzbaren Speichermedium.

[0053] In der Verarbeitungseinheit 206 wird ein Betriebssystem ausgeführt. Das Betriebssystem koordiniert verschiedene Komponenten innerhalb des Datenverarbeitungssystems 200 aus **Fig. 2** und stellt deren Steuerung bereit. Das Betriebssystem kann ein handelsübliches Betriebssystem für jegliche Art von Datenverarbeitungsplattform sein, darunter Server-Systeme, Personal Computer und mobile Einheiten, ohne darauf beschränkt zu sein. Ein objektorientiertes oder anderweitiges Programmiersystem kann gemeinsam mit dem Betriebssystem ausgeführt werden und aus Programmen oder Anwendungen heraus, die in dem Datenverarbeitungssystem 200 ausgeführt werden, Aufrufe an das Betriebssystem bereitstellen.

[0054] Anweisungen für das Betriebssystem, das objektorientierte Programmiersystem und Anwendungen oder Programme wie z.B. die Anwendung 105 aus **Fig. 1** befinden sich auf Speichereinheiten, z.B. in Gestalt von Code 226A auf dem Festplattenlaufwerk 226, und können zur Ausführung durch die Verarbeitungseinheit 206 in mindestens einen des einen oder der mehreren Arbeitsspeicher wie z.B. den Hauptspeicher 208 geladen werden. Die Prozesse für die veranschaulichenden Ausführungs-

formen können durch die Verarbeitungseinheit 206 unter Verwendung von computerrealisierten Anweisungen durchgeführt werden, die sich in einem Arbeitsspeicher wie zum Beispiel dem Hauptarbeitspeicher 208, dem Nur-Lese-Speicher 224 oder in einer oder mehreren Peripherie-Einheiten befinden können.

[0055] Darüber hinaus kann in einem Fall der Code 226A über ein Netzwerk 201A von einem entfernt angeordneten System 201B heruntergeladen werden, wo ein ähnlicher Code 201C in einer Speichereinheit 201D gespeichert wird. In einem weiteren Fall kann der Code 226A über das Netzwerk 201A auf das entfernt angeordnete System 201B heruntergeladen werden, wo der heruntergeladene Code 201C in einer Speichereinheit 201D gespeichert wird.

[0056] Die Hardware aus den **Fig. 1** und **Fig. 2** kann je nach Realisierung variieren. Zusätzlich zu oder anstatt der in den **Fig. 1** und **Fig. 2** dargestellten Hardware können auch andere interne Hardware- oder Peripherie-Einheiten wie z.B. ein Flash-Arbeitspeicher, ein gleichwertiger nicht flüchtiger Arbeitsspeicher oder optische Plattenlaufwerke und dergleichen verwendet werden. Zusätzlich hierzu können die Prozesse der veranschaulichenden Ausführungsformen auf ein Mehrprozessor-Datenverarbeitungssystem angewendet werden.

[0057] Bei manchen veranschaulichenden Beispielen kann das Datenverarbeitungssystem 200 ein persönlicher digitaler Assistent (PDA) sein, der im Allgemeinen mit einem Flash-Arbeitspeicher konfiguriert wird, um nicht flüchtigen Arbeitsspeicher zum Speichern von Betriebssystemdateien und/oder durch einen Benutzer erzeugten Daten bereitzustellen. Ein Bussystem kann einen oder mehrere Busse aufweisen, wie z.B. einen E/A-Bus und einen PCI-Bus. Selbstverständlich lässt sich das Bussystem mit jeglicher Art von Datenübertragungsstruktur oder -architektur realisieren, die eine Übertragung von Daten zwischen verschiedenen Komponenten oder Einheiten vorsieht, die mit der Struktur oder Architektur verbunden werden.

[0058] Eine Datenübertragungseinheit kann eine oder mehrere Einheiten enthalten, die zum Senden und Empfangen von Daten verwendet werden, z.B. einen Modem oder einen Netzwerkadapter. Ein Arbeitsspeicher kann zum Beispiel der Hauptarbeitspeicher 208 oder ein Cache sein, wie z.B. der in dem North Bridge & Memory Controller Hub 202 befindliche Cache. Eine Verarbeitungseinheit kann einen oder mehrere Prozessoren oder eine oder mehrere CPUs enthalten.

[0059] Die dargestellten Beispiele aus den **Fig. 1** und **Fig. 2** und die oben beschriebenen Beispiele sollen keine architektonische Beschränkung bedeuten.

Zum Beispiel kann das Datenverarbeitungssystem 200 zusätzlich dazu, dass es in Gestalt einer mobilen oder am Körper tragbaren Einheit vorliegen kann, auch ein Tablet Computer, ein Laptop Computer oder eine Telefoneinheit sein.

[0060] Wenn ein Computer oder Datenverarbeitungssystem als eine virtuelle Maschine, eine virtuelle Einheit oder eine virtuelle Komponente beschrieben wird, arbeitet die virtuelle Maschine, die virtuelle Einheit oder die virtuelle Komponente in der Art des Datenverarbeitungssystems 200 unter Verwendung einer virtualisierten Ausprägung einiger oder aller in dem Datenverarbeitungssystem 200 dargestellten Komponenten. Zum Beispiel ist in einer virtuellen Maschine, virtuellen Einheit oder virtuellen Komponente die Verarbeitungseinheit 206 als eine virtualisierte Instanz aller oder einer bestimmten Anzahl von Hardware-Verarbeitungseinheiten 206 ausgeprägt, die in einem Host-Datenverarbeitungssystem verfügbar sind, der Hauptarbeitspeicher 208 ist als eine virtualisierte Instanz des gesamten oder eines Teils des Hauptarbeitspeichers 208 ausgeprägt, der in dem Host-Datenverarbeitungssystem verfügbar sein kann, und die Festplatte 226 ist als eine virtualisierte Instanz der gesamten oder eines Teils der Festplatte 226 ausgeprägt, die in dem Host-Datenverarbeitungssystem verfügbar sein kann. Das Host-Datenverarbeitungssystem wird in diesen Fällen durch das Datenverarbeitungssystem 200 dargestellt.

[0061] Unter Bezugnahme auf **Fig. 3** stellt diese Figur ein Blockschaubild einer beispielhaften Datenstrom-Anwendungsumgebung 300 gemäß einer veranschaulichenden Ausführungsform dar. Bei einer Ausführungsform wird die Datenstrom-Anwendungsumgebung 300 konfiguriert, um eine Datenstromanwendung auszuführen, die ein Beispiel für die Anwendung 105A/105B aus **Fig. 1** ist.

[0062] Bei der veranschaulichten Ausführungsform enthält die Datenstrom-Anwendungsumgebung 300 ein Verwaltungssystem 305 und eine Mehrzahl von Datenverarbeitungsknoten 310A bis 310D (hierin allgemein als Knoten 310 bezeichnet), d.h. Hosts, die unter Verwendung eines oder mehrerer Datenübertragungsnetzwerke 320 im Datenaustausch miteinander verbunden werden. Bei einer Ausführungsform enthält das Datenübertragungsnetzwerk 320 einen oder mehrere Server, ein oder mehrere Netzwerke oder eine oder mehrere Datenbanken und verwendet ein Datenübertragungsprotokoll, um Daten zwischen den Datenverarbeitungsknoten 310 zu übertragen. Bei einer Ausführungsform wird ein Datenbanksystem 315, das eine Datenbank 340 enthält, mit dem Netzwerk 320 im Datenaustausch verbunden, um eine Datenübertragung mit dem Datenverarbeitungsknoten 310 zu ermöglichen. Bei einer bestimmten Ausführungsform ist das Datenbanksystem 315 ein

Beispiel für den Server 106 mit der Datenbank 109 aus **Fig. 1**, und die Knoten 310A bis 310D sind Beispiele für das Datenverarbeitungssystem 104, die Clients 110, 112 und 114 und die Einheit 132 aus **Fig. 1**.

[0063] Bei der veranschaulichten Ausführungsform werden der eine oder die mehreren Datenverarbeitungsknoten 310A bis 310D wie für den Datenverarbeitungsknoten 310D gezeigt konfiguriert, sodass jeder der Datenverarbeitungsknoten 310A bis 310D ein oder mehrere PEs 312 aufweist und jedes PE einen oder mehrere Operatoren 314 enthält. Die Operatoren 314 sind grundlegende Bausteine der Datenstromanwendung und ermöglichen, dass die Datenstromanwendung eine verteilte Datenverarbeitungsanwendung ist. Bei manchen Ausführungsformen führt jeder der Datenverarbeitungsknoten 310A bis 310D ein Linux-Betriebssystem aus, und jedes PE 312 steht für einen Linux-Prozess, und innerhalb eines jeden Linux-Prozesses gibt es Operatoren 314, die einen oder mehrere Aspekte der Datenstromanwendungen ausführen, sodass die Datenstromanwendung auf die Operatoren 314, die PEs 312 und die Datenverarbeitungsknoten 310A bis 310D verteilt wird.

[0064] Bei manchen Ausführungsformen enthält das Verwaltungssystem 305 einen Operatorgraphen 332 und einen Datenstromverwalter 334. Bei manchen Ausführungsformen schreiben Entwickler unter Verwendung des Stream-Computing-Dienstes Datenstromanwendungen, indem sie den Operatorgraphen 332 definieren, der kompiliert und an das Verwaltungssystem 305 übergeben wird.

[0065] Der Datenstromverwalter 334 platziert die Operatoren in dem Operatorgraphen 332 in einem oder mehreren PEs in einem oder mehreren Datenverarbeitungsknoten, die einen oder mehrere der Datenverarbeitungsknoten 310A bis 310D enthalten können. Bei manchen Ausführungsformen enthält der Operatorgraph 332 eine Beschreibung der Datenstrom-Netzwerktopologie, z.B. den Speicherort, die Anordnung, die Verbindungen und die Funktionalität von verschiedenen Operatoren in den Knoten 310 der Datenstrom-Anwendungsumgebung 300.

[0066] PEs wie z.B. das PE 312 hosten einen oder mehrere Operatoren, z.B. den bzw. die Operatoren 314, die eingehende Tupel verarbeiten und Ausgabe-Tupel erzeugen. Ein PE hat eine unabhängige Verarbeitungseinheit und wird in einem Knoten ausgeführt, z.B. einem beliebigen der Datenverarbeitungsknoten 310A bis 310D. In einer Datenstromanwendung werden die PEs 312 miteinander verbunden, sodass Daten von einem PE zum nächsten fließen. Daten fließen in Form eines Tupels von einem zu einem weiteren PE, wobei das Tupel eine

Abfolge von einem oder mehreren Attributen hat, die einer Entität zugehörig sind.

[0067] Bei manchen Ausführungsformen überwacht der Datenstromverwalter 334 die PEs 312 auf einen Leistungsabfall und ersetzt ein PE mit einem Leistungsabfall vor seinem Ausfall durch eine Kopie seiner selbst mit einer besseren Leistung. Streaming-Anwendungen verarbeiten potenziell enorme Mengen an Daten in Echtzeit. Wenn eine Streaming-Anwendung bei ihrer Verarbeitung „in Verzug gerät“, kann die von ihr durchgeführte Analyse ihren Wert einbüßen, d.h. Ergebnisse der Analyse beruhen auf den vergangenen, nicht auf den neuesten Daten. Eine Anwendung kann einfach deswegen in Verzug geraten, weil ein Teil der Anwendung, z.B. ein einziges PE, eine mangelnde Leistung aufweist. Somit stellt ein Erkennen und Ersetzen eines PEs mit einem Leistungsabfall vor dessen Ausfall, wie hierin beschrieben, beträchtliche Verbesserungen hinsichtlich der Leistung und Effizienz von Streaming-Anwendungen bereit.

[0068] Unter Bezugnahme auf **Fig. 4** stellt diese Figur ein Blockschaubild eines beispielhaften Datenverarbeitungsknotens 400 gemäß einer veranschaulichenden Ausführungsform dar. Bei einer Ausführungsform ist der Datenverarbeitungsknoten 400 ein Beispiel für die Datenverarbeitungsknoten 310A bis 310D aus **Fig. 3**.

[0069] Bei der veranschaulichten Ausführungsform enthält der Datenverarbeitungsknoten 400 einen oder mehrere Prozessoren (CPUs) 402, eine Netzwerkschnittstelle 408, eine Zwischenverbindung 410, einen Arbeitsspeicher 412 und einen Speicher 414. Der Datenverarbeitungsknoten 400 kann auch eine E/A-Einheitenschnittstelle 404 enthalten, die verwendet wird, um E/A-Einheiten 406 wie z.B. Tastatur-, Anzeige- und Mauseinheiten mit dem Datenverarbeitungsknoten 400 zu verbinden. Bei manchen Ausführungsformen wird die hierin beschriebene Funktionalität auf eine Mehrzahl von Systemen verteilt, die Kombinationen von Systemen auf Software- und/oder Hardware-Grundlage enthalten kann, zum Beispiel anwendungsspezifische integrierte Schaltungen (Application-Specific Integrated Circuits, ASICs), Computerprogramme oder Smartphone-Anwendungen.

[0070] Bei manchen Ausführungsformen ruft jede CPU 402 Programmieranweisungen ab, die in dem Arbeitsspeicher 412 oder dem Speicher 414 gespeichert werden, und führt sie aus. Entsprechend speichert bei manchen Ausführungsformen die CPU 402 Anwendungsdaten, die sich in dem Arbeitsspeicher 412 befinden, und ruft sie ab. Bei manchen Ausführungsformen überträgt die Zwischenverbindung 410 Programmieranweisungen und Anwendungsdaten zwischen jeder CPU 402, jeder E/A-Einheitenschnitt-

stelle 404, jedem Speicher 414, jeder Netzwerkschnittstelle 408 und jedem Arbeitsspeicher 412.

[0071] Bei manchen Ausführungsformen werden ein oder mehrere PEs 416 in dem Arbeitsspeicher 412 gespeichert. Bei manchen derartigen Ausführungsformen enthält ein PE 416 einen oder mehrere Datenstromoperatoren 418. Bei manchen Ausführungsformen wird ein PE 416 zugewiesen, um durch nur eine einzige CPU 402 ausgeführt zu werden, obwohl bei anderen Ausführungsformen die Datenstromoperatoren 418 eines PEs 416 einen oder mehrere Threads enthalten, die in zwei oder mehr der CPUs 402 ausgeführt werden. Der Arbeitsspeicher 412 ist im Allgemeinen in Form eines Direktzugriffsspeichers enthalten, z.B. in Form eines statischen Direktzugriffsspeichers (Static Random Access Memory, SRAM), eines dynamischen Direktzugriffsspeichers (Dynamic Random Access Memory, DRAM) oder eines Flash-Speichers. Der Speicher 414 enthält einen Puffer 422 und ist im Allgemeinen in Form eines nicht flüchtigen Arbeitsspeichers enthalten, z.B. in Form eines Festplattenlaufwerks, einer Halbleitereinheit (Solid State Device, SSD) oder entfernbaren Arbeitsspeicherkarten, eines optischen Speichers, Flash-Arbeitsspeichereinheiten, eines netzwerkverbundenen Speichers (Network Attached Storage, NAS) oder Verbindungen mit Speicherbereichsnetzwerk-Einheiten (Storage Area Network, SAN) oder anderer Einheiten, die nicht flüchtige Daten speichern können.

[0072] Bei manchen Ausführungsformen enthält ein Datenstromverwalter einen oder mehrere der Datenstromoperatoren 418, die zu einem PE 416 kompiliert werden. Bei manchen Ausführungsformen enthält der Arbeitsspeicher 412 zwei oder mehr der PEs 416, wobei jedes PE einen oder mehrere der Datenstromoperatoren 418 hat. Bei manchen Ausführungsformen enthält jeder Datenstromoperator 418 einen Teil eines Codes, der Tupeln verarbeitet, die in ein PE fließen, und Tupel an andere Datenstromoperatoren 418 in demselben PE, in anderen PEs oder sowohl in demselben als auch in anderen PEs in einer Stream-Computing-Anwendung ausgibt. Bei manchen Ausführungsformen leiten die PEs 416 Tupel an andere PEs weiter, die sich in demselben Datenverarbeitungsknoten 400 oder in anderen Datenverarbeitungsknoten binden, auf die über Datenübertragungsnetzwerke zugegriffen werden kann. Zum Beispiel gibt bei einer Ausführungsform ein erstes PE 416 in einem ersten Datenverarbeitungsknoten 400 Tupel an ein zweites PE 416 in einem zweiten Datenverarbeitungsknoten 400 aus.

[0073] Bei manchen Ausführungsformen enthalten die PEs 416 einen Leistungsüberwacher 420. Bei manchen Ausführungsformen überwacht der Leistungsüberwacher 420 das PE 416 auf einen Leistungsabfall und gibt Benachrichtigungssignale oder

-daten aus, wenn ein Leistungsabfall des PEs 416 erkannt wird. Zum Beispiel sendet der Leistungsüberwacher 420 bei manchen Ausführungsformen ein Signal an einen Datenstromverwalter (z.B. den Datenstromverwalter 334 aus **Fig. 3**), wenn der Leistungsüberwacher 420 einen Leistungsabfall des PEs 416 erkennt.

[0074] Bei manchen Ausführungsformen erkennt ein Leistungsüberwacher 420 einen Leistungsabfall, indem er das PE 416 auf ein Eintreten einer Bedingung überwacht, die als ein Leistungsabfallindikator dient, der sich von einem PE 416 zu einem weiteren PE 416 unterscheiden kann. Bei manchen Ausführungsformen können sich die Schwellenwerte für Leistungsabfallindikator-Bedingungen von einem PE 416 zu einem weiteren PE 416 unterscheiden. Somit ist bei manchen derartigen Ausführungsformen eine Leistungsabfallindikator-Bedingung für jedes PE 416 konfigurierbar. Bei manchen Ausführungsformen enthält eine Leistungsabfallindikator-Bedingung Ausführungsbedingungen, die dem PE 416 zugehörig sind und von einem vordefinierten oder erwarteten Muster abweichen. Zum Beispiel enthalten bei manchen Ausführungsformen Indikatoren für einen Leistungsabfall eine anomale Anzahl von Tupeln in Warteschlangen (d.h. Tupeln, die an Eingabeanschlüssen von Operatoren in dem PE darauf warten, verarbeitet zu werden), eine anomale Arbeitsspeichernutzung, Anzahl und Arten von verarbeiteten Ausnahmen sowie eine anomale CPU-Auslastung. Bei manchen Ausführungsformen überwacht der Leistungsüberwacher 420 das PE 416 und verwaltet Statistiken, die für die normale oder erwartete Leistung des PEs 416 stehen. Bei manchen Ausführungsformen protokolliert ein Leistungsüberwacher 420 die Nutzung derartiger Datenverarbeitungsressourcen durch das PE 416, wodurch normale Nutzungsmuster erstellt werden können und wodurch zudem anomale Nutzungsmuster erkannt werden können. Bei manchen Ausführungsformen enthält eine anomale Anzahl von Tupeln in Warteschlangen eine Anzahl von Tupeln in Warteschlangen, die höher als normal ist. Bei manchen Ausführungsformen enthält eine anomale Arbeitsspeichernutzung anomal hohe Änderungsraten bei der Arbeitsspeichernutzung. Bei manchen Ausführungsformen enthält eine anomale CPU-Auslastung anomal hohe Änderungsraten bei der CPU-Auslastung.

[0075] Unter Bezugnahme auf **Fig. 5** stellt diese Figur ein Blockschaubild eines beispielhaften Verwaltungssystems 500 gemäß einer veranschaulichenden Ausführungsform dar. Bei einer Ausführungsform ist das Verwaltungssystem 500 ein Beispiel für das Verwaltungssystem 305 aus **Fig. 3**.

[0076] Bei der veranschaulichten Ausführungsform enthält das Verwaltungssystem 500 einen oder meh-

rere Prozessoren (CPUs) 502, eine Netzwerkschnittstelle 508, eine Zwischenverbindung 510, einen Arbeitsspeicher 512 und einen Speicher 514. Das Verwaltungssystem 500 kann auch eine E/A-Einheitschnittstelle 504 enthalten, die E/A-Einheiten 506 wie z.B. Tastatur-, Anzeige- und Mauseinheiten mit dem Verwaltungssystem 500 verbindet. Bei manchen Ausführungsformen wird die hierin beschriebene Funktionalität auf eine Mehrzahl von Systemen verteilt, die Kombinationen von Systemen auf Software- und/oder Hardware-Grundlage enthalten kann, zum Beispiel anwendungsspezifische integrierte Schaltungen (ASICs), Computerprogramme oder Smartphone-Anwendungen.

[0077] Bei manchen Ausführungsformen ruft jede CPU 502 Programmieranweisungen ab, die in dem Arbeitsspeicher 512 oder dem Speicher 514 gespeichert werden, und führt sie aus. Entsprechend speichert bei manchen Ausführungsformen jede CPU 502 Anwendungsdaten, die sich in dem Arbeitsspeicher 512 oder dem Speicher 514 befinden, und ruft sie ab. Die Zwischenverbindung 510 dient zum Verschieben von Daten wie z.B. Programmieranweisungen und Anwendungsdaten zwischen der CPU 502, der E/A-Einheitschnittstelle 504, der Speichereinheit 514, der Netzwerkschnittstelle 508 und dem Arbeitsspeicher 512. Der Speicher 514 ist im Allgemeinen enthalten, um für einen nicht flüchtigen Arbeitsspeicher wie z.B. ein Festplattenlaufwerk, eine Halbleitereinheit (SSD), entfernbare Speicherkarten, einen optischen Speicher, Flash-Arbeitsspeichereinheiten, einen netzwerkverbundenen Speicher (NAS), Verbindungen mit Speicherbereichsnetzwerk-Einheiten (SAN) oder die Cloud repräsentativ zu sein. Bei manchen Ausführungsformen wird die Netzwerkschnittstelle 508 konfiguriert, um Daten über das Datenübertragungsnetzwerk zu übertragen.

[0078] Bei manchen Ausführungsformen speichert der Arbeitsspeicher 512 einen Datenstromverwalter 516. Zusätzlich speichert der Speicher 514 bei manchen solchen Ausführungsformen einen Operatorgraphen 522, der definiert, wie Tupel zur Verarbeitung an PEs weitergeleitet werden. Bei der veranschaulichten Ausführungsform enthält der Datenstromverwalter 516 außerdem ein Analysemodul 518 für ein vorläufiges PE und einen Einplaner 524. Bei der veranschaulichten Ausführungsform enthält das Analysemodul 518 für ein vorläufiges PE einen Leistungsüberwacher 520.

[0079] Bei manchen Ausführungsformen überwacht das Analysemodul 518 für ein vorläufiges PE die PEs auf einen Leistungsabfall und ersetzt ein PE mit einem Leistungsabfall vor seinem Ausfall durch eine Kopie seiner selbst mit einer besseren Leistung. Streaming-Anwendungen verarbeiten potenziell enorme Mengen an Daten in Echtzeit. Wenn eine

Streaming-Anwendung bei ihrer Verarbeitung „in Verzug gerät“, kann die von ihr durchgeführte Analyse ihren Wert einbüßen, d.h. Ergebnisse der Analyse beruhen auf den vergangenen, nicht auf den neuesten Daten. Eine Anwendung kann einfach deswegen in Verzug geraten, weil ein Teil der Anwendung, z.B. ein einziges PE, eine mangelnde Leistung aufweist. Somit stellt ein Erkennen und Ersetzen eines PEs mit einem Leistungsabfall vor dessen Ausfall, wie hierin beschrieben, beträchtliche Verbesserungen hinsichtlich der Leistung und Effizienz von Streaming-Anwendungen bereit.

[0080] Bei manchen Ausführungsformen überwacht der Leistungsüberwacher 520 eine Streaming-Anwendung, indem er Statistiken verwaltet, die für die Leistung der verschiedenen PEs stehen, aus denen die Streaming-Anwendung besteht. Bei manchen Ausführungsformen ist der Leistungsüberwacher 520, wie gezeigt, ein Teil des Datenstromverwalters 516. Bei anderen Ausführungsformen wird der Leistungsüberwacher 520 während der Laufzeit aufgerufen, kann jedoch eine von dem Laufzeit-Datenstromverwalter 516 getrennte Entität sein.

[0081] Bei manchen Ausführungsformen überwacht der Leistungsüberwacher 520 PEs auf einen Leistungsabfall und gibt Benachrichtigungssignale oder -daten aus, wenn ein Leistungsabfall eines oder mehrerer PEs erkannt wird. Zum Beispiel sendet der Leistungsüberwacher 520 bei manchen Ausführungsformen ein Signal an den Datenstromverwalter 516, wenn der Leistungsüberwacher 520 einen Leistungsabfall eines PEs erkennt.

[0082] Bei manchen Ausführungsformen erkennt ein Leistungsüberwacher 520 einen Leistungsabfall, indem er PEs auf ein Eintreten einer Bedingung überwacht, die als ein Leistungsabfallindikator dient, der sich von einem PE 416 zu einem weiteren PE 416 unterscheiden kann. Bei manchen Ausführungsformen können sich die Schwellenwerte für Leistungsabfallindikator-Bedingungen von einem PE 416 zu einem weiteren PE 416 unterscheiden. Somit ist bei manchen derartigen Ausführungsformen eine Leistungsabfallindikator-Bedingung für jedes PE 416 konfigurierbar. Bei manchen Ausführungsformen enthält eine Leistungsabfallindikator-Bedingung Ausführungsbedingungen, die einem PE zugehörig sind und von einem vordefinierten oder erwarteten Muster abweichen. Zum Beispiel enthalten bei manchen Ausführungsformen Indikatoren für einen Leistungsabfall eine anomale Anzahl von Tupeln in Warteschlangen (d.h. Tupeln, die an Eingabeanschlüssen von Operatoren in dem PE darauf warten, verarbeitet zu werden), eine anomale Arbeitsspeichernutzung, Anzahl und Arten von verarbeiteten Ausnahmen sowie eine anomale CPU-Auslastung. Bei manchen Ausführungsformen überwacht der Leistungsüberwacher 520 PEs und

verwaltet Statistiken, die für die normale oder erwartete Leistung der PEs stehen. Bei manchen Ausführungsformen protokolliert ein Leistungsüberwacher 520 die Nutzung derartiger Datenverarbeitungsressourcen durch PEs, wodurch normale Nutzungsmuster erstellt werden können und wodurch zudem anomale Nutzungsmuster erkannt werden können. Bei manchen Ausführungsformen enthält eine anomale Anzahl von Tupeln in Warteschlangen eine Anzahl von Tupeln in Warteschlangen, die höher als normal ist. Bei manchen Ausführungsformen enthält eine anomale Arbeitsspeichernutzung anomal hohe Änderungsraten bei der Arbeitsspeichernutzung. Bei manchen Ausführungsformen enthält eine anomale CPU-Auslastung anomal hohe Änderungsraten bei der CPU-Auslastung.

[0083] Bei manchen Ausführungsformen analysiert das Analysemodul 518 ein PE, das durch den Leistungsüberwacher 520 als ein PE identifiziert wird, das ein oder mehrere Anzeichen für einen Leistungsabfall aufweist. Bei manchen Ausführungsformen analysiert das Analysemodul 518 das PE, um zu ermitteln, ob die durch den Leistungsüberwacher 520 identifizierten Anzeichen tatsächlich Anzeichen für ein anderes Problem als ein Leistungsabfall des PEs sind, z.B. für einen Rückstau oder eine oder mehrere Cluster-Datenverarbeitungsressourcen, die ihre Kapazitätsgrenzen erreichen. Wenn das Analysemodul 518 kein anderes Problem als einen Leistungsabfall des PEs erkennt, evaluiert das Analysemodul 518 bei manchen derartigen Ausführungsformen als Nächstes das PE unter Verwendung eines vorläufigen PEs. Bei manchen derartigen Ausführungsformen tauscht der Datenstromverwalter mit einem Einplaner Daten aus, um einen Knoten zum Hosten des vorläufigen PEs 524 zu identifizieren. Bei manchen Ausführungsformen 524 sucht der Einplaner nach einem anderen Knoten als dem Knoten, der das PE hostet. Bei manchen Ausführungsformen sucht der Einplaner 524 auch auf Grundlage von vorgegebenen Regeln oder Kriterien nach einem Knoten, zum Beispiel einer Verfügbarkeit von Datenverarbeitungsressourcen und einer Nähe zu vor- und nachgelagerten PEs.

[0084] Sobald der Einplaner 524 einen Host-Knoten identifiziert, startet das Analysemodul 518 bei manchen Ausführungsformen das vorläufige PE in dem identifizierten Knoten während der Laufzeit, während zugleich eine Laufzeitoperation des PEs beibehalten wird. Bei manchen derartigen Ausführungsformen wird das vorläufige PE mit demselben vorgelagerten PE wie das analysierte PE verbunden, sodass sowohl das vorläufige PE als auch das analysierte PE zur selben Zeit denselben Datenstrom von Tupeln von demselben vorgelagerten PE empfangen. Bei manchen derartigen Ausführungsformen führen sowohl das vorläufige PE als auch das analysierte PE denselben definierten Prozess für den

Datenstrom von Tupeln durch. Bei manchen derartigen Ausführungsformen führt das analysierte PE den definierten Prozess für den Datenstrom von Tupeln durch, was zu einem Ausgabedatenstrom von Tupeln führt, den das PE an ein nachgelagertes PE überträgt, während die Ausgabe des vorläufigen PEs mit keinerlei nachgelagerten Elementen verbunden wird.

[0085] Bei manchen Ausführungsformen vergleicht das Analysemodul 518 eine Operation des analysierten PEs mit einer Operation des vorläufigen PEs auf Grundlage einer Leistungsmetrik, die einer oder mehreren Datenverarbeitungsressourcen zugehörig ist. Wenn das Analysemodul 518 ermittelt, dass das vorläufige PE eine bessere Leistung als das analysierte PE aufweist, ersetzt das Analysemodul 518 bei manchen derartigen Ausführungsformen das analysierte PE durch das vorläufige PE, zum Beispiel, indem die Ausgabe des vorläufigen PEs anstatt des analysierten PEs mit dem nachgelagerten PE verbunden wird, woraufhin das analysierte PE beendet wird.

[0086] Unter Bezugnahme auf **Fig. 6** stellt diese Figur ein Blockschaubild eines beispielhaften Analysemoduls 600 für ein vorläufiges PE gemäß einer veranschaulichenden Ausführungsform dar. Bei einer Ausführungsform ist ein Analysemodul 600 für ein vorläufiges PE ein Beispiel für das Analysemodul 518 für ein vorläufiges PE aus **Fig. 5**.

[0087] Bei der veranschaulichten Ausführungsform enthält das Analysemodul 600 für ein vorläufiges PE einen Leistungsüberwacher 602, ein Anfangsanalysemodul 604 und einen Fehlerbeheber 606 für ein vorläufiges PE. Bei der veranschaulichten Ausführungsform enthält der Fehlerbeheber 606 für ein vorläufiges PE einen Erzeuger 610 für ein vorläufiges PE, einen Leistungsvergleicher 612 und einen Systemaktualisierer 614. Bei manchen Ausführungsformen wird die hierin beschriebene Funktionalität auf eine Mehrzahl von Systemen verteilt, die Kombinationen von Systemen auf Software- und/oder Hardware-Grundlage enthalten kann, zum Beispiel anwendungsspezifische integrierte Schaltungen (ASICs), Computerprogramme oder Smartphone-Anwendungen.

[0088] Bei der veranschaulichten Ausführungsform überwacht der Leistungsüberwacher 602 eine Streaming-Anwendung, indem er Statistiken verwaltet, die für die Leistung der verschiedenen PEs stehen, aus denen die Streaming-Anwendung besteht. Bei manchen Ausführungsformen überwacht der Leistungsüberwacher 602 PEs auf einen Leistungsabfall und gibt Benachrichtigungssignale oder -daten aus, wenn ein Leistungsabfall eines oder mehrerer PEs erkannt wird. Zum Beispiel sendet der Leistungsüberwacher 602 bei manchen Ausführungsformen

ein Signal an das Anfangsanalysemodul 604, wenn der Leistungsüberwacher 602 einen Leistungsabfall eines PEs erkennt, z.B. eines in **Fig. 6** gezeigten PEs 608 mit einem Leistungsabfall.

[0089] Bei manchen Ausführungsformen erkennt der Leistungsüberwacher 602 einen Leistungsabfall des PEs 608 mit einem Leistungsabfall, indem er das PE 608 mit einem Leistungsabfall auf ein Eintreten einer Bedingung überwacht, die als ein Leistungsabfallindikator dient, der sich von einem PE zu einem weiteren PE unterscheiden kann. Bei manchen Ausführungsformen können sich auch die Schwellenwerte für Leistungsabfallindikator-Bedingungen von einem PE zu einem weiteren PE unterscheiden. Bei manchen derartigen Ausführungsformen ist eine Leistungsabfallindikator-Bedingung somit für jedes PE konfigurierbar. Bei manchen Ausführungsformen enthält eine Leistungsabfallindikator-Bedingung Ausführungsbedingungen, die einem PE zugehörig sind und von einem vordefinierten oder erwarteten Muster abweichen. Zum Beispiel enthalten bei manchen Ausführungsformen Indikatoren für einen Leistungsabfall eine anomale Anzahl von Tupeln in Warteschlangen (d.h. Tupeln, die an Eingabeanschlüssen von Operatoren in dem PE darauf warten, verarbeitet zu werden), eine anomale Arbeitsspeichernutzung, Anzahl und Arten von verarbeiteten Ausnahmen sowie eine anomale CPU-Auslastung. Bei manchen Ausführungsformen überwacht der Leistungsüberwacher 602 eine Mehrzahl von PEs, z.B. das PE 608 mit einem Leistungsabfall. Bei alternativen Ausführungsformen überwacht der Leistungsüberwacher 602 nur das PE 608 mit einem Leistungsabfall. Bei manchen Ausführungsformen verwaltet der Leistungsüberwacher 602 Statistiken, die für die normale oder erwartete Leistung des PEs 608 mit einem Leistungsabfall vor einem Leistungsabfall des PEs 608 stehen. Bei manchen Ausführungsformen protokolliert der Leistungsüberwacher 602 die Nutzung derartiger Datenverarbeitungsressourcen durch das PE 608 mit einem Leistungsabfall, wodurch normale Nutzungsmuster erstellt werden können und wodurch zudem anomale Nutzungsmuster erkannt werden können. Bei manchen Ausführungsformen enthält eine anomale Anzahl von Tupeln in Warteschlangen eine Anzahl von Tupeln in Warteschlangen, die höher als normal ist. Bei manchen Ausführungsformen enthält eine anomale Arbeitsspeichernutzung anomal hohe Änderungsraten bei der Arbeitsspeichernutzung. Bei manchen Ausführungsformen enthält eine anomale CPU-Auslastung anomal hohe Änderungsraten bei der CPU-Auslastung.

[0090] Bei manchen Ausführungsformen analysiert das Anfangsanalysemodul 604 das PE 608 mit einem Leistungsabfall, wenn es durch den Leistungsüberwacher 602 als ein PE identifiziert wird, das ein oder mehrere Anzeichen für einen Leistungs-

abfall aufweist. Bei manchen Ausführungsformen analysiert das Anfangsanalysemodul 604 das PE 608 mit einem Leistungsabfall, um zu ermitteln, ob die durch den Leistungsüberwacher 602 identifizierten Anzeichen tatsächlich Anzeichen für ein anderes Problem als ein Leistungsabfall des PEs 608 mit einem Leistungsabfall sind, z.B. für einen Rückstau oder eine oder mehrere Cluster-Datenverarbeitungsressourcen, die ihre Kapazitätsgrenzen erreichen.

[0091] Wenn das Anfangsanalysemodul 602 kein anderes Problem als einen Leistungsabfall des PEs 608 mit einem Leistungsabfall erkennt, benachrichtigt das Anfangsanalysemodul 604 bei manchen derartigen Ausführungsformen den vorläufigen Fehlerbeheber 606, der als Nächstes das PE 608 mit einem Leistungsabfall unter Verwendung eines vorläufigen PEs 616 evaluiert. Bei manchen derartigen Ausführungsformen tauscht der Fehlerbeheber 606 für ein vorläufiges PE mit einem Einplaner (z.B. dem Einplaner 524 aus **Fig. 5**) Daten aus, um einen Knoten zum Hosten des vorläufigen PEs 616 zu identifizieren. Bei manchen Ausführungsformen sucht der Einplaner nach einem anderen Knoten als dem Knoten, der das PE hostet. Bei manchen Ausführungsformen sucht der Einplaner auch auf Grundlage von vorgegebenen Regeln oder Kriterien nach einem Knoten, zum Beispiel einer Verfügbarkeit von Datenverarbeitungsressourcen und einer Nähe zu vor- und nachgelagerten PEs. Sobald der Einplaner einen Host-Knoten identifiziert, benachrichtigt der Einplaner bei manchen Ausführungsformen den Erzeuger 610 für ein vorläufiges PE. Der Erzeuger 610 für das vorläufige PE startet das vorläufige PE 616 in dem identifizierten Knoten während der Laufzeit, während eine Laufzeitoperation des PEs 608 mit einem Leistungsabfall beibehalten wird. Bei manchen derartigen Ausführungsformen verbindet der Erzeuger 610 für ein vorläufiges PE eine vorgelagerte Seite des vorläufigen PEs 616 mit demselben vorgelagerten PE wie das PE 608 mit einem Leistungsabfall, sodass sowohl das vorläufige PE 616 als auch das PE 608 mit einem Leistungsabfall zur selben Zeit denselben Datenstrom von Tupeln von demselben vorgelagerten PE empfangen. Bei manchen derartigen Ausführungsformen führen sowohl das vorläufige PE 616 als auch das PE 608 mit einem Leistungsabfall denselben definierten Prozess für den Datenstrom von Tupeln durch, wodurch der Leistungsvergleicher 612 die Leistung des PEs 608 mit einem Leistungsabfall mit derjenigen des vorläufigen PEs 616 vergleichen kann. Bei manchen derartigen Ausführungsformen führt das PE 608 mit einem Leistungsabfall den definierten Prozess für den Datenstrom von Tupeln durch, was zu einem Ausgabedatenstrom von Tupeln führt, den das PE 608 mit einem Leistungsabfall an ein nachgelagertes PE 608 überträgt, während die Ausgabe des vorläufigen PEs 616 mit keinerlei nachgelagerten Elementen verbunden wird.

[0092] Bei manchen Ausführungsformen vergleicht der Leistungsvergleicher 612 die Operation des PEs mit einem Leistungsabfall mit derjenigen des vorläufigen PEs auf Grundlage einer Leistungsmetrik, die einer oder mehreren Datenverarbeitungsressourcen zugehörig ist. Zum Beispiel vergleicht der Leistungsvergleicher 612 bei manchen Ausführungsformen die Operation des PEs mit einem Leistungsabfall mit derjenigen des vorläufigen PEs 616 indem er eines oder mehrere einer CPU-Auslastung, einer Arbeitsspeichernutzung und einer Geschwindigkeit des Tupeldurchsatzes vergleicht. Wenn der Leistungsvergleicher 612 ermittelt, dass das vorläufige PE 616 eine bessere Leistung als das PE 608 mit einem Leistungsabfall aufweist, benachrichtigt der Leistungsvergleicher 612 bei manchen derartigen Ausführungsformen den Systemaktualisierer 614. Wenn als Reaktion auf die Benachrichtigung durch den Leistungsvergleicher 612, dass das vorläufige PE 616 eine bessere Leistung als das PE 608 mit einem Leistungsabfall aufweist, ersetzt der Systemaktualisierer 614 das PE 608 mit einem Leistungsabfall durch das vorläufige PE 616, zum Beispiel, indem die Ausgabe des vorläufigen PEs 616 anstatt des PEs 608 mit einem Leistungsabfall mit dem nachgelagerten PE verbunden wird, und der Systemaktualisierer 614 beendet das PE 608 mit einem Leistungsabfall. Bei manchen Ausführungsformen sendet der Systemaktualisierer 614 über eine Benutzerschnittstelle 618 eine Benachrichtigung über den Ersatz des PEs 608 mit einem Leistungsabfall durch das vorläufige PE 616 an den Benutzer.

[0093] Unter Bezugnahme auf **Fig. 7** stellt diese Figur ein Blockschaubild eines Operatorgraphen 700 für eine Stream-Computing-Anwendung gemäß einer veranschaulichenden Ausführungsform dar. Der in **Fig. 7** gezeigte Operatorgraph wird aus Gründen der Klarheit als ein nicht beschränkendes Beispiel für eine Streaming-Anwendung gezeigt, die von offenbaren Ausführungsformen profitieren kann.

[0094] Bei der veranschaulichten Ausführungsform ist der Operatorgraph 700 ein Graph für eine Stream-Computing-Anwendung, der bei einer oder mehreren Quellen 710 beginnt und bei einer oder mehreren Senken 712, 714 endet. Dieser Fluss von Quelle zu Senke kann hierin im Allgemeinen auch als ein Ausführungspfad bezeichnet werden. Obwohl **Fig. 7** eine Abstraktion darstellt, um verbundene PEs PE1 bis PE10 zu zeigen, kann der Operatorgraph 700 Datenflüsse zwischen Datenstromoperatoren (z.B. dem Operator 314 aus **Fig. 3**) innerhalb derselben oder verschiedenen PEs enthalten. In der Regel empfangen PEs Tupel von dem Datenstrom und geben Tupel als Ausgabe in den Datenstrom ein (mit Ausnahme einer Senke, wo der Datenstrom endet, oder einer Quelle, wo der Datenstrom beginnt).

[0095] Bei der veranschaulichten Ausführungsform enthält der Operatorgraph 700 zehn PEs (als PE1 bis PE10 gekennzeichnet), die in den Datenverarbeitungsknoten 702, 704, 706 und 708 ausgeführt werden. Bei manchen Ausführungsformen enthält ein PE einen oder mehrere Datenstromoperatoren, die miteinander verbunden werden, um einen unabhängig ausgeführten Prozess mit seiner eigenen Prozess-ID (PID) und seinem eigenen Arbeitsspeicherbereich zu bilden. Bei Ausführungsformen, bei denen zwei (oder mehr) PEs unabhängig ausgeführt werden, kann eine Interprozess-Datenübertragung unter Verwendung eines „Transports“, z.B. eines Netzwerk-Sockets, eines TCP/IP-Sockets oder eines gemeinsam genutzten Arbeitsspeichers stattfinden. Wenn jedoch Datenstromoperatoren miteinander verbunden werden, können die verbundenen Datenstromoperatoren schnellere Datenübertragungsmethoden verwenden, um Tupel zwischen Datenstromoperatoren in jedem PE weiterzuleiten.

[0096] Der Operatorgraph 700 beginnt bei einer Quelle 710 und endet bei einer Senke 712, 714. Der Datenverarbeitungsknoten 702 enthält die PEs PE1, PE2 und PE3. Die Quelle 710 fließt in das PE PE1, das wiederum Tupel ausgibt, die durch das PE2 und das PE3 empfangen werden. So kann das PE1 zum Beispiel in einem Tupel empfangene Datenattribute aufteilen und manche Datenattribute in einem neuen Tupel an das PE2 weiterleiten, während andere Datenattribute in einem weiteren neuen Tupel an das PE3 weitergeleitet werden. In einem zweiten Beispiel kann das PE1 manche empfangenen Tupel an das PE2 weiterleiten, während andere Tupel an das PE3 weitergeleitet werden. Tupel, die an das PE2 fließen, werden durch die in dem PE2 enthaltenen Datenstromoperatoren verarbeitet, und die resultierenden Tupel werden dann an ein PE4 in dem Datenverarbeitungsknoten 704 ausgegeben. Entsprechend fließen die durch das PE4 ausgegebenen Tupel in die Operatorsenke PE6 712. Auf ähnliche Weise gelangen Tupel, die von dem PE3 zu einem PE5 fließen, ebenfalls zu den Operatoren in dem Senken-PE6 712. Zusätzlich zur Funktion als Senke für diesen beispielhaften Operatorgraphen kann das PE6 somit konfiguriert werden, um eine gemeinsame Operation durchzuführen, wobei Tupel kombiniert werden, die von dem PE4 und dem PE5 empfangen werden. Dieser beispielhafte Operatorgraph zeigt außerdem Tupel, die von dem PE3 zu einem PE7 in dem Datenverarbeitungsknoten 706 fließen, der wiederum Tupel zeigt, die zu einem PE8 fließen und zu dem PE7 zurückgeschleift werden. Von dem PE8 ausgegebene Tupel fließen zu einem PE9 in dem Datenverarbeitungsknoten 708, der wiederum Tupel ausgibt, die durch Operatoren in einem Senken-PE verarbeitet werden sollen, zum Beispiel dem PE10 714.

[0097] Bei manchen Ausführungsformen wird ein durch ein bestimmtes PE empfangenes Tupel im Allgemeinen nicht als dasselbe Tupel betrachtet, das im weiteren Verlauf ausgegeben wird, da das Ausgabe-Tupel auf eine wie auch immer geartete Weise geändert wird. Zum Beispiel wird bei manchen Ausführungsformen ein Attribut hinzugefügt, gelöscht oder geändert bzw. Metadaten werden hinzugefügt, gelöscht oder geändert. Allerdings enthalten manche Ausführungsformen ein PE, das ein Ausgabe-Tupel gegenüber dem Eingabe-Tupel auf keinerlei Weise ändert. Im Allgemeinen kann ein bestimmtes Tupel, das durch ein PE ausgegeben wird, auch dann nicht als dasselbe Tupel wie ein entsprechendes Eingabe-Tupel betrachtet werden, wenn das Eingabe-Tupel durch das PE nicht geändert wird. Um jedoch die vorliegende Beschreibung und die Ansprüche zu vereinfachen, kann ein Ausgabe-Tupel, das dieselben Datenattribute wie ein entsprechendes Eingabe-Tupel hat, hierin als dasselbe Tupel bezeichnet werden. In der veranschaulichten Ausführungsform wird ein Leistungsmonitor, wie hierin beschrieben, konfiguriert, um eines oder mehrere der PEs PE1 bis PE10 auf einen Leistungsabfall zu überwachen.

[0098] Unter Bezugnahme auf **Fig. 8** stellt diese Figur ein Blockschaubild eines Operatorgraphen 800 für eine Stream-Computing-Anwendung mit einem vorläufigen PE gemäß einer veranschaulichenden Ausführungsform dar. Der in **Fig. 8** gezeigte Operatorgraph wird aus Gründen der Klarheit als ein nicht beschränkendes Beispiel einer abgewandelten Version des Operatorgraphen 700 aus **Fig. 7** gezeigt, der bei einer Quelle 810 beginnt und bei einer Senke 812, 814 endet und von offenbarten Ausführungsformen profitieren kann.

[0099] In dem veranschaulichten Beispiel erkennt ein Leistungsmonitor (z.B. der Leistungsmonitor 520 aus **Fig. 5** oder der Leistungsmonitor 602 aus **Fig. 6**) einen Leistungsabfall des PE2s, indem er das PE2 überwacht und erkennt, dass Ausführungsbedingungen, die dem PE2 zugehörig sind, von einem vordefinierten oder erwarteten Muster abweichen. Zum Beispiel erkennt der Leistungsüberwacher bei manchen Ausführungsformen Indikatoren für einen Leistungsabfall wie z.B. eine anomale Anzahl von Tupeln in Warteschlangen (d.h. Tupel, die an Eingabeanschlüssen von Operatoren in dem PE darauf warten, verarbeitet zu werden), eine anomale Arbeitsspeichernutzung, Anzahl und Arten von verarbeiteten Ausnahmen und/oder eine anomale CPU-Auslastung. In dem veranschaulichten Beispiel wurde kein anderes Problem als ein Leistungsabfall erkannt, sodass ein vorläufiges PE (in **Fig. 8** als PPE bezeichnet) in einem anderen Knoten gestartet wird. Da sich das PE2 in einem Knoten 802 befindet, befindet sich das PPE in einem Knoten 804, hätte jedoch alternativ auch in einem Knoten 806 oder einem Knoten 808 gestartet werden können.

[0100] Das vorläufige PE wird während der Laufzeit gestartet, während eine Laufzeitoperation des PE2s beibehalten wird. Zum Zwecke eines Leistungsvergleichs zwischen dem vorläufigen PE und dem PE2 wird das vorläufige PE anfänglich nur auf seiner vorgelagerten Seite über eine Kante 816 mit dem PE1 verbunden, sodass sowohl das vorläufige PE als auch das PE2 zur selben Zeit denselben Datenstrom von Tupeln von demselben vorgelagerten PE1 empfangen und beide denselben definierten Prozess für den Datenstrom von Tupeln durchführen, was einen Vergleich der Leistung des vorläufigen PEs mit derjenigen des PE2s ermöglicht.

[0101] In dem veranschaulichten Beispiel vergleicht ein Leistungsvergleicher oder Datenstromverwalter die Operation des vorläufigen PEs mit derjenigen des PE2s auf Grundlage einer Leistungsmetrik, die einer oder mehreren Datenverarbeitungsressourcen zugehörig ist. Zum Beispiel werden bei manchen Ausführungsformen das vorläufige PE und das PE2 auf Grundlage von einem oder mehreren einer CPU-Auslastung, einer Arbeitsspeichernutzung und/oder einer Geschwindigkeit eines Tupeldurchsatzes verglichen. Wenn das vorläufige PE eine bessere Leistung als das PE2 aufweist, ersetzt bei manchen derartigen Ausführungsformen das vorläufige PE das PE2, zum Beispiel, indem die Ausgabe-/ nachgelagerte Seite des vorläufigen PEs anstatt des PE2s mit dem nachgelagerten PE4 des PE2s verbunden wird, zum Beispiel, indem eine Kante 818 von dem vorläufigen PE mit dem PE4 verbunden wird und das PE2 beendet wird.

[0102] Unter Bezugnahme auf **Fig. 9** stellt diese Figur einen Ablaufplan eines beispielhaften dynamischen PE-Ersetzungsprozesses 900 gemäß einer veranschaulichenden Ausführungsform dar. Bei einer bestimmten Ausführungsform führt das PE-Analysemodul 600 aus **Fig. 6** den Prozess 900 durch.

[0103] Bei einer Ausführungsform startet der Prozess 900 in einem Block 902 eine Datenstromanwendung. Als Nächstes überwacht der Prozess 900 in einem Block 904 Verarbeitungselemente auf einen Leistungsabfall. Als Nächstes prüft der Prozess 900 in einem Block 906, ob bei irgendwelchen Verarbeitungselementen ein Leistungsabfall erkannt wurde. Wenn dies nicht der Fall ist, kehrt der Prozess 900 zu dem Block 904 zurück, um mit dem Überwachen fortzufahren. Andernfalls fährt der Prozess 900 mit einem Block 908 fort, in dem der Prozess 900 eine Evaluierung des PEs mit einem Leistungsabfall beginnt. In einem Block 908 prüft der Prozess 900 das PE mit einem Leistungsabfall auf ein Anzeichen dafür, dass ein Rückstau die Ursache für den Leistungsabfall ist, gefolgt von einem Block 912, in dem der Prozess 900 die Funktionsniveaus von Datenverarbeitungsressourcen des Clusters von Knoten prüft,

in dem das PE mit einem Leistungsabfall bereitgestellt wird. Wenn der Prozess 900 einen Rückstau oder einen Mangel an verfügbaren oder ausreichenden Cluster-Datenverarbeitungsressourcen als die wahrscheinliche Ursache für den Leistungsabfall des PEs mit einem Leistungsabfall erkennt, fährt der Prozess mit einem Block 910 fort, in dem der Prozess andere nach dem Stand der Technik bekannte Abhilfemaßnahmen anwendet, um den Rückstau und/oder den Mangel an Cluster-Datenverarbeitungsressourcen zu beheben. Andernfalls fährt der Prozess mit einem Block 914 fort, in dem der Prozess 900 einen Fehlerbehebungsprozess für ein vorläufiges PE durchführt, zum Beispiel gemäß dem Prozess 1000 aus **Fig. 10**, dem Prozess 1100 aus **Fig. 11**, dem Prozess 1200 aus **Fig. 12** oder einem Prozess, der Teile von zwei oder mehr der Prozesse 1000, 1100, und/oder 1200 kombiniert.

[0104] Nach dem Fehlerbehebungsprozess für ein vorläufiges PE in dem Block 914 kehrt der Prozess 900 zu dem Block 904 zurück, um mit dem Überwachen von Verarbeitungselementen auf einen Leistungsabfall fortzufahren. Bei der veranschaulichten Ausführungsform wird das Überwachen fortgesetzt, bis ein weiteres PE mit einem Leistungsabfall festgestellt wird oder bis die Datenstromanwendung stoppt, wie in einem Block 916 kenntlich gemacht wird, wonach der Prozess 900 endet.

[0105] Unter Bezugnahme auf **Fig. 10** stellt diese Figur einen Ablaufplan eines beispielhaften dynamischen PE-Ersetzungsprozesses 1000 gemäß einer veranschaulichenden Ausführungsform dar. Bei einer bestimmten Ausführungsform führt das PE-Analysemodul 600 aus **Fig. 6** den Prozess 1000 durch. Bei einer Ausführungsform ist der Prozess 1000 ein Beispiel für den Fehlerbehebungsprozess für ein vorläufiges PE in dem Block 914 aus **Fig. 9**.

[0106] Bei der veranschaulichten Ausführungsform testet der Prozess 1000 die Verwendung eines vorläufigen Verarbeitungselements als ein möglicher Ersatz für ein PE mit einem Leistungsabfall. In einem Block 1002 ermittelt der Prozess 1000 einen Speicherort für ein vorläufiges PE. Bei manchen Ausführungsformen verwendet der Prozess 1000 einen Einplaner, um einen Knoten zu identifizieren, in dem das vorläufige PE bereitgestellt werden soll. Bei manchen derartigen Ausführungsformen ermittelt der Einplaner einen besten Knoten zum Ausführen des vorläufigen PEs. Bei manchen derartigen Ausführungsformen schließt der Einplaner den Knoten aus, in dem das PE mit einem Leistungsabfall ausgeführt wird, wenn er einen besten Knoten zum Ausführen des vorläufigen PEs ermittelt.

[0107] Als Nächstes startet der Prozess 1000 in einem Block 1004 das vorläufige PE in dem Knoten, der in dem Block 1002 ausgewählt wird. Bei man-

chen Ausführungsformen wird das vorläufige PE während der Laufzeit gestartet, während eine Laufzeitoperation des PEs mit einem Leistungsabfall beibehalten wird. Bei manchen Ausführungsformen wird das vorläufige PE nur auf seiner vorgelagerten Seite verbunden und wird mit dem PE verbunden, das dem PE mit einem Leistungsabfall unmittelbar vorgelagert ist. Diese Verbindung führt dazu, dass sowohl das vorläufige PE als auch das PE mit einem Leistungsabfall zur selben Zeit denselben Datenstrom von Tupeln von demselben vorgelagerten PE empfangen.

[0108] Als Nächstes vergleicht der Prozess 1000 in einem Block 1006 die Leistung des PEs mit einem Leistungsabfall und des vorläufigen PEs. Bei manchen derartigen Ausführungsformen führt das vorläufige PE dieselbe definierte Verarbeitung für den Datenstrom von Tupeln wie das PE mit einem Leistungsabfall durch, was einen auf der Leistung beruhenden Vergleich des vorläufigen PEs mit dem PE mit einem Leistungsabfall ermöglicht. Bei manchen Ausführungsformen vergleicht ein Leistungsvergleicher oder Datenstromverwalter die Operation des vorläufigen PEs mit derjenigen des PEs mit einem Leistungsabfall auf Grundlage einer Leistungsmetrik, die einer oder mehreren Datenverarbeitungsressourcen zugehörig ist. Zum Beispiel werden bei manchen Ausführungsformen das vorläufige PE und das PE mit einem Leistungsabfall auf Grundlage von einem oder mehreren einer CPU-Auslastung, einer Arbeitsspeichernutzung und/oder einer Geschwindigkeit eines Tupeldurchsatzes verglichen.

[0109] Als Nächstes ermittelt der Prozess 1000 in einem Block 1008, ob die Leistung des vorläufigen PEs besser als die Leistung des PEs mit einem Leistungsabfall ist. Wenn das vorläufige PE eine bessere Leistung als das PE mit einem Leistungsabfall aufweist, führt bei manchen derartigen Ausführungsformen der Prozess 1000 in einem Block 1010 eine Umschaltung auf das vorläufige PE durch, um das PE mit einem Leistungsabfall zu ersetzen, und das PE mit einem Leistungsabfall wird beendet. Bei manchen Ausführungsformen enthält die Umschaltung ein Verbinden der Ausgabe-/nachgelagerten Seite des vorläufigen PEs mit dem PE, das dem PE mit einem Leistungsabfall unmittelbar nachgelagert ist.

[0110] Unter Bezugnahme auf **Fig. 11** stellt diese Figur einen Ablaufplan eines beispielhaften dynamischen PE-Ersetzungsprozesses 1100 gemäß einer veranschaulichenden Ausführungsform dar. Bei einer bestimmten Ausführungsform führt das PE-Analysemodul 600 aus **Fig. 6** den Prozess 1100 durch. Bei einer Ausführungsform ist der Prozess 1100 ein Beispiel für den Fehlerbehebungsprozess für ein vorläufiges PE in dem Block 914 aus **Fig. 9**.

[0111] Bei der veranschaulichten Ausführungsform testet der Prozess 1100 die Verwendung eines aus einer Mehrzahl von vorläufigen Verarbeitungselementen als möglichen Ersatz für ein PE mit einem Leistungsabfall. Bei manchen Ausführungsformen ist die Anzahl von vorläufigen PEs ein fester, vorgegebener Wert. Bei manchen Ausführungsformen ist die Anzahl von vorläufigen PEs ein durch einen Benutzer konfigurierbarer Wert, der anfänglich auf einen Standardwert gesetzt wird und durch Benutzereingaben anpassbar ist. Bei manchen Ausführungsformen ermittelt der Prozess 1100 die Anzahl von zu verwendenden vorläufigen PEs auf Grundlage einer beliebigen Anzahl von realisierungsspezifischen Faktoren, zum Beispiel der Art eines replizierten PEs mit einem Leistungsabfall (d.h. der Operationen, die durch einen oder mehrere Operatoren für das PE mit einem Leistungsabfall durchgeführt werden), der Größe der Datenstromanwendung, der Anzahl von verfügbaren Knoten, die bereits durch die Datenstromanwendung verwendet werden, und/oder beliebiger anderer gewünschter Faktoren.

[0112] In einem Block 1102 ermittelt der Prozess 1100 Speicherorte für jedes aus der Mehrzahl von vorläufigen PEs. Bei manchen Ausführungsformen verwendet der Prozess 1100 einen Einplaner, um Knoten zu identifizieren, in denen vorläufige PEs bereitgestellt werden sollen. Bei manchen derartigen Ausführungsformen ermittelt der Einplaner eine Liste von besten Knoten zum Ausführen der vorläufigen PEs. Bei manchen Ausführungsformen ermittelt der Einplaner außerdem die Anzahl von bereitzustellenden vorläufigen PEs, zum Beispiel auf Grundlage einer Verfügbarkeit von Knoten, in denen die vorläufigen PEs bereitgestellt werden sollen. Bei manchen derartigen Ausführungsformen schließt der Einplaner den Knoten aus, in dem das PE mit einem Leistungsabfall ausgeführt wird, wenn er beste Knoten zum Ausführen der vorläufigen PEs ermittelt.

[0113] Als Nächstes startet der Prozess 1100 in einem Block 1104 die vorläufigen PEs in dem betreffenden Knoten, der in dem Block 1102 ausgewählt wird. Bei manchen Ausführungsformen werden die vorläufigen PEs während der Laufzeit gestartet, während eine Laufzeitoperation des PEs mit einem Leistungsabfall beibehalten wird. Bei manchen Ausführungsformen werden die vorläufigen PEs nur auf ihrer vorgelagerten Seite verbunden und werden alle mit dem PE verbunden, das dem PE mit einem Leistungsabfall unmittelbar vorgelagert ist. Diese Verbindung führt dazu, dass sowohl das PE mit einem Leistungsabfall als auch der Satz von vorläufigen PEs zur selben Zeit denselben Datenstrom von Tupeln von demselben vorgelagerten PE empfangen.

[0114] Als Nächstes vergleicht der Prozess 1100 in einem Block 1106 die Leistung des PEs mit einem

Leistungsabfall und der vorläufigen PEs. Bei manchen derartigen Ausführungsformen führen die vorläufige PEs dieselbe definierte Verarbeitung für den Datenstrom von Tupeln wie das PE mit einem Leistungsabfall durch, was einen auf der Leistung beruhenden Vergleich des PEs mit einem Leistungsabfall mit den vorläufigen PEs ermöglicht. Bei manchen Ausführungsformen vergleicht ein Leistungsvergleicher oder Datenstromverwalter die Operation des PEs mit einem Leistungsabfall mit derjenigen der vorläufigen PEs mit einem Leistungsabfall auf Grundlage einer Leistungsmetrik, die einer oder mehreren Datenverarbeitungsressourcen zugehörig ist. Zum Beispiel werden bei manchen Ausführungsformen die vorläufigen PEs und das PE mit einem Leistungsabfall auf Grundlage von einem oder mehreren einer CPU-Auslastung, einer Arbeitsspeichernutzung und/oder einer Geschwindigkeit eines Tupeldurchsatzes verglichen.

[0115] Als Nächstes ermittelt der Prozess 1100 in einem Block 1108, ob die Leistung irgendeines der vorläufigen PEs besser als die Leistung des PEs mit einem Leistungsabfall ist. Wenn irgendeines der vorläufigen PEs eine bessere Leistung als das PE mit einem Leistungsabfall aufweist, führt bei manchen derartigen Ausführungsformen der Prozess 1100 in einem Block 1110 eine Umschaltung auf das vorläufige PE mit der besten Leistung durch, um das PE mit einem Leistungsabfall zu ersetzen, und die verbleibenden vorläufigen PEs werden beendet. Bei manchen Ausführungsformen enthält die Umschaltung ein Verbinden der Ausgabe-/nachgelagerten Seite des vorläufigen PEs mit dem PE, das dem PE mit einem Leistungsabfall unmittelbar nachgelagert ist.

[0116] Unter Bezugnahme auf **Fig. 12** stellt diese Figur einen Ablaufplan eines beispielhaften dynamischen PE-Ersetzungsprozesses 1200 gemäß einer veranschaulichenden Ausführungsform dar. Bei einer bestimmten Ausführungsform führt das PE-Analysemodul 600 aus **Fig. 6** den Prozess 1200 durch. Bei einer Ausführungsform ist der Prozess 1200 ein Beispiel für den Fehlerbehebungsprozess für ein vorläufiges PE in dem Block 914 aus **Fig. 9**.

[0117] Bei der veranschaulichten Ausführungsform testet der Prozess 1200 die Verwendung eines vorläufigen Verarbeitungselements als ein möglicher Ersatz für ein PE mit einem Leistungsabfall. In einem Block 1202 ermittelt der Prozess 1200 einen Speicherort für ein vorläufiges PE. Bei manchen Ausführungsformen verwendet der Prozess 1200 einen Einplaner, um einen Knoten zu identifizieren, in dem das vorläufige PE bereitgestellt werden soll. Bei manchen derartigen Ausführungsformen ermittelt der Einplaner eine Liste von besten Knoten zum Ausführen der vorläufigen PEs. Bei manchen Ausführungsformen ermittelt der Einplaner außerdem die

Anzahl von bereitzustellenden vorläufigen PEs, zum Beispiel auf Grundlage einer Verfügbarkeit von Knoten, in denen die vorläufigen PEs bereitgestellt werden sollen. Bei manchen derartigen Ausführungsformen schließt der Einplaner den Knoten aus, in dem das PE mit einem Leistungsabfall ausgeführt wird, wenn er den bzw. die besten Knoten zum Ausführen der vorläufigen PEs ermittelt.

[0118] Als Nächstes startet der Prozess 1200 in einem Block 1204 das vorläufige PE in dem Knoten, der in dem Block 1202 ausgewählt wird. Bei manchen Ausführungsformen wird das vorläufige PE während der Laufzeit gestartet, während eine Laufzeitoperation des PEs mit einem Leistungsabfall beibehalten wird. Bei manchen Ausführungsformen wird das vorläufige PE nur auf seiner vorgelagerten Seite verbunden und wird mit dem PE verbunden, das dem PE mit einem Leistungsabfall unmittelbar vorgelagert ist. Diese Verbindung führt dazu, dass sowohl das vorläufige PE als auch das PE mit einem Leistungsabfall zur selben Zeit denselben Datenstrom von Tupeln von demselben vorgelagerten PE empfangen.

[0119] Als Nächstes vergleicht der Prozess 1200 in einem Block 1206 die Leistung des PEs mit einem Leistungsabfall und des vorläufigen PEs. Bei manchen derartigen Ausführungsformen führt das vorläufige PE dieselbe definierte Verarbeitung für den Datenstrom von Tupeln wie das PE mit einem Leistungsabfall durch, was einen auf der Leistung beruhenden Vergleich des vorläufigen PEs mit dem PE mit einem Leistungsabfall ermöglicht. Bei manchen Ausführungsformen vergleicht ein Leistungsvergleicher oder Datenstromverwalter die Operation des vorläufigen PEs mit derjenigen des PEs mit einem Leistungsabfall auf Grundlage einer Leistungsmetrik, die einer oder mehreren Datenverarbeitungsressourcen zugehörig ist. Zum Beispiel werden bei manchen Ausführungsformen das vorläufige PE und das PE mit einem Leistungsabfall auf Grundlage von einem oder mehreren einer CPU-Auslastung, einer Arbeitsspeichernutzung und/oder einer Geschwindigkeit eines Tupeldurchsatzes verglichen.

[0120] Als Nächstes ermittelt der Prozess 1200 in einem Block 1208, ob die Leistung des vorläufigen PEs besser als die Leistung des PEs mit einem Leistungsabfall ist. Wenn das vorläufige PE keine bessere Leistung als das PE mit einem Leistungsabfall aufweist, ermittelt der Prozess 1200 bei manchen derartigen Ausführungsformen in einem Block 1212, ob eine maximale Anzahl von vorläufigen PEs ausprobiert wurde. Wenn dies der Fall ist, endet der Prozess. Wenn dies nicht der Fall ist, kehrt der Prozess zu dem Block 1202 zurück, um unter Verwendung eines weiteren vorläufigen PEs an einem anderen Speicherort (das z.B. in einem anderen Knoten als dem Knoten bereitgestellt wird, in dem das PE mit

einem Leistungsabfall bereitgestellt wird, und auch in einem anderen Knoten als der bzw. die Knoten, die für frühere Iterationen der Blöcke 1202 bis 1208 verwendet wurden) eine nächste Iteration der Blöcke 1202 bis 1208 durchzuführen.

[0121] Bei manchen Ausführungsformen ist die maximale Anzahl von vorläufigen PEs in dem Block 1212 ein fester, vorgegebener Wert. Bei manchen Ausführungsformen ist die maximale Anzahl von vorläufigen PEs in dem Block 1212 ein durch einen Benutzer konfigurierbarer Wert, der anfänglich auf einen Standardwert gesetzt wird und durch Benutzereingaben anpassbar ist. Bei manchen Ausführungsformen ermittelt der Prozess 1100 in dem Block 1212 die maximale Anzahl von vorläufigen PEs auf Grundlage einer beliebigen Anzahl von realisierungsspezifischen Faktoren, zum Beispiel der Art eines replizierten PEs mit einem Leistungsabfall (d.h. der Operationen, die durch einen oder mehrere Operatoren für das PE mit einem Leistungsabfall durchgeführt werden), der Größe der Datenstromanwendung, der Anzahl von verfügbaren Knoten, die bereits durch die Datenstromanwendung verwendet werden, und/oder beliebiger anderer gewünschter Faktoren.

[0122] Wenn eine beliebige der Iterationen der Blöcke 1202 bis 1208 dazu führt, dass das vorläufige PE eine bessere Leistung als das PE mit einem Leistungsabfall aufweist, führt der Prozess 1200 bei manchen derartigen Ausführungsformen in dem Block 1210 eine Umschaltung auf das vorläufige PE durch, um das PE mit einem Leistungsabfall zu ersetzen, und das PE mit einem Leistungsabfall wird beendet. Bei manchen Ausführungsformen enthält die Umschaltung ein Verbinden der Ausgabe-/nachgelagerten Seite des vorläufigen PEs mit dem PE, das dem PE mit einem Leistungsabfall unmittelbar nachgelagert ist.

[0123] Für die Auslegung der Ansprüche und der Beschreibung sind die folgenden Definitionen und Abkürzungen zu verwenden. Wie hierin verwendet, sollen die Begriffe „weist auf“, „aufweisend“, „enthält“, „enthaltend“, „hat“, „mit“, „fasst“ oder „fassend“ bzw. jegliche andere Abwandlung hiervon einen nicht ausschließlichen Einschluss ausdrücken. Zum Beispiel ist eine Zusammensetzung, eine Mischung, ein Prozess, ein Verfahren, ein Gegenstand oder eine Vorrichtung, die bzw. der eine Liste von Elementen aufweist, nicht notwendigerweise auf diese Elemente beschränkt, sondern kann auch andere Elemente enthalten, die nicht ausdrücklich aufgeführt werden oder einer solchen Zusammensetzung, Mischung, einem solchen Prozessor, Verfahren, Artikel oder einer solchen Vorrichtung innewohnen.

[0124] Zusätzlich wird der Begriff „veranschaulichend“ hierin im Sinne von „als Beispiel dienend,

exemplarisch oder zur Veranschaulichung dienend“ verwendet. Jede hierin als „veranschaulichend“ beschriebene Ausführungsform oder Ausgestaltung ist nicht zwingend so zu verstehen, als sei sie gegenüber anderen Ausführungsformen oder Entwürfen bevorzugt oder vorteilhaft. Die Begriffe „mindestens einer“ und „einer oder mehrere“ werden verstanden, als enthielten sie jede Ganzzahl größer als oder gleich eins, d.h. eins, zwei, drei, vier usw. Der Begriff „eine Mehrzahl“ wird verstanden, als enthielte er jede Ganzzahl größer als oder gleich zwei, d.h. zwei, drei, vier, fünf usw. Der Begriff „Verbindung“ kann sowohl eine indirekte „Verbindung“ als auch eine direkte „Verbindung“ enthalten.

[0125] Wenn in der Patentschrift auf „eine bestimmte Ausführungsform“, „eine Ausführungsform“, „eine Beispielausführungsform“ usw. Bezug genommen wird, bedeutet dies, dass die beschriebene Ausführungsform ein bestimmtes Merkmal, eine bestimmte Struktur oder ein bestimmtes kennzeichnendes Element enthalten kann, wobei jedoch nicht notwendigerweise jede Ausführungsform das betreffende Merkmal, die betreffende Struktur oder das betreffende kennzeichnende Element enthält. Zudem beziehen sich solche Formulierungen nicht notwendigerweise auf dieselbe Ausführungsform. Wenn in Verbindung mit einer Ausführungsform ein bestimmtes Merkmal, eine Struktur oder ein kennzeichnendes Element beschrieben wird, wird des Weiteren angenommen, dass der Fachmann weiß, wie sich ein solches Merkmal, eine Struktur oder ein kennzeichnendes Element in Verbindung mit anderen Ausführungsformen bewerkstelligen lassen würde, unabhängig davon, ob es bzw. sie explizit beschrieben ist.

[0126] Die Begriffe „circa“, „im Wesentlichen“, „ungefähr“ und Abwandlungen hiervon sollen den Grad an Fehlerhaftigkeit enthalten, der auf Grundlage der zum Zeitpunkt der Anmeldungseinreichung verfügbaren Ausstattung einer Messung der betreffenden Menge zugehörig ist. So kann „circa“ zum Beispiel einen Bereich von $\pm 8\%$ oder 5% oder 2% eines bestimmten Werts enthalten.

[0127] 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 beschriebenen Ausführungsformen zu ermöglichen.

[0128] Somit werden in den veranschaulichenden Ausführungsformen ein computerrealisiertes Verfahren, System bzw. eine computerrealisierte Vorrichtung sowie ein Computerprogrammprodukt zum Verwalten einer Teilnahme an Online-Gemeinschaften und andere damit zusammenhängende Merkmale, Funktionen bzw. Operationen bereitgestellt. Wo eine Ausführungsform oder ein Teil davon in Bezug auf eine Art von Einheit beschrieben wird, werden das computerrealisierte Verfahren, System bzw. die computerrealisierte Vorrichtung, das Computerprogrammprodukt oder ein Teil davon für eine Verwendung mit einer geeigneten und vergleichbaren Ausprägung dieser Art von Einheit angepasst oder konfiguriert.

[0129] Wo eine Ausführungsform als in einer Anwendung realisiert beschrieben wird, gilt die Bereitstellung der Anwendung in einem SaaS-Modell (Software as a Service) als innerhalb des inhaltlichen Umfangs der veranschaulichenden Ausführungsformen liegend. In einem SaaS-Modell wird die Fähigkeit der eine Ausführungsform realisierenden Anwendung einem Benutzer bereitgestellt, indem die Anwendung in einer Cloud-Infrastruktur ausgeführt wird. Der Benutzer kann unter Verwendung einer Vielfalt von Client-Einheiten über eine Thin-Client-Schnittstelle wie z.B. einen Web-Browser (z.B. eine eMail-Nachricht auf Web-Grundlage) oder andere leichtgewichtige Client-Anwendungen auf die Anwendung zugreifen. Der Benutzer verwaltet bzw. steuert die zugrundeliegende Cloud-Infrastruktur nicht, darunter das Netzwerk, die Server, die Betriebssysteme oder den Speicher der Cloud-Infrastruktur. In manchen Fällen kann der Benutzer nicht einmal die Fähigkeiten der SaaS-Anwendung verwalten oder steuern. In manchen anderen Fällen kann die SaaS-Realisierung der Anwendung eine mögliche Ausnahme in Gestalt von eingeschränkten, benutzerspezifischen Anwendungskonfigurationseinstellungen gestatten.

[0130] 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.

[0131] 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 Kombination daraus. Ein computerlesbares Speichermedium soll in der Verwendung hierin nicht als transitorische 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. Lichtwellenleiterkabel durchlaufende Lichtimpulse) oder durch einen Draht übertragene elektrische Signale.

[0132] 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 Netzwerkkartentypkarte 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.

[0133] 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 externen 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.

[0134] 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.

[0135] 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.

[0136] 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.

[0137] 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.

[0138] Ausführungsformen der vorliegenden Erfindung können auch als Teil einer Dienstvereinbarung mit einem Kundenunternehmen, einer gemeinnützigen Organisation, einer Regierungsstelle, einer inter-

nen Organisationsstruktur oder Ähnlichem bereitgestellt werden. Aspekte dieser Ausführungsformen können ein Konfigurieren eines Computersystems, um einige oder alle der hier beschriebenen Verfahren durchzuführen, und ein Bereitstellen von Software, Hardware und Web-Diensten enthalten, die einige oder alle der hier beschriebenen Verfahren realisieren. Aspekte dieser Ausführungsformen können außerdem ein Analysieren der Arbeitsabläufe des Kunden, ein Erzeugen von Empfehlungen als Reaktion auf die Analyse, ein Schaffen von Systemen, die Teile der Empfehlungen realisieren, ein Integrieren der Systeme in bestehende Prozesse und Infrastrukturen, ein Messen der Verwendung der Systeme, ein Zuordnen von Aufwendungen zu Nutzern der Systeme und ein In-Rechnung-Stellen für eine Verwendung der Systeme enthalten. Obwohl die obigen Ausführungsformen der vorliegenden Erfindung jeweils beschrieben wurden, indem ihre jeweiligen Vorteile aufgeführt wurden, ist die vorliegende Erfindung nicht auf eine bestimmte Kombination hiervon beschränkt. Vielmehr können solche Ausführungsformen gemäß der beabsichtigten Bereitstellung der vorliegenden Erfindung auf eine beliebige Weise und Anzahl kombiniert werden, ohne ihre vorteilhaften Wirkungen einzubüßen.

Patentansprüche

1. Computerrealisiertes Verfahren, aufweisend: Überwachen einer verteilten Datenverarbeitungsanwendung während der Laufzeit auf ein Eintreten einer Bedingung; wobei die verteilte Datenverarbeitungsanwendung einen ersten Knoten aufweist, der ein Verarbeitungselement (Processing Element, PE) enthält, das einen ersten Datenstrom von Tupeln von einem vorgelagerten PE empfängt, einen definierten Prozess für den ersten Datenstrom von Tupeln durchführt, der zu einem zweiten Datenstrom von Tupeln führt, und den zweiten Datenstrom von Tupeln an ein nachgelagertes PE überträgt, und wobei die Bedingung ein Eintreten einer Leistungsabfallbedingung für das PE enthält; Starten eines ersten vorläufigen PEs in einem zweiten Knoten während der Laufzeit, während eine Laufzeitoperation des PEs beibehalten wird, wobei das erste vorläufige PE den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfängt und den definierten Prozess für den ersten Datenstrom von Tupeln durchführt; Vergleichen einer Operation des PEs mit einer Operation des ersten vorläufigen PEs auf Grundlage einer Leistungsmetrik, die einer Datenverarbeitungsressource zugehörig ist; und Verbinden des ersten vorläufigen PEs, um den zweiten Datenstrom von Tupeln von dem PE zu dem nachgelagerten PE durch einen dritten Datenstrom von Tupeln von dem ersten vorläufigen PE zu dem nachgelagerten PE zu ersetzen.

2. Computerrealisiertes Verfahren nach Anspruch 1, des Weiteren aufweisend ein Auswählen des zweiten Knotens aus einer Mehrzahl von Knoten zum Ausführen des ersten vorläufigen PEs.

3. Computerrealisiertes Verfahren nach Anspruch 2, wobei das Auswählen des zweiten Knotens aus der Mehrzahl von Knoten zumindest teilweise auf einer Verfügbarkeit von Datenverarbeitungsressourcen in dem zweiten Knoten beruht.

4. Computerrealisiertes Verfahren nach Anspruch 1, wobei das erste vorläufige PE den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfängt und den definierten Prozess für den ersten Datenstrom von Tupeln durchführt, während das PE den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfängt und den definierten Prozess für den ersten Datenstrom von Tupeln durchführt.

5. Computerrealisiertes Verfahren nach Anspruch 4, des Weiteren aufweisend ein Starten eines zweiten vorläufigen PEs in einem dritten Knoten während der Laufzeit, während eine Laufzeitoperation des PEs beibehalten wird, wobei das zweite vorläufige PE den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfängt und den definierten Prozess für den ersten Datenstrom von Tupeln durchführt, während das erste vorläufige PE und das PE jeweils den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfangen und jeweils den definierten Prozess für den ersten Datenstrom von Tupeln durchführen.

6. Computerrealisiertes Verfahren nach Anspruch 5, wobei das Vergleichen der Operation des PEs mit der Operation des ersten vorläufigen PEs des Weiteren ein Vergleichen der Operation des zweiten vorläufigen PEs mit der Operation des PEs und mit der Operation des ersten vorläufigen PEs auf Grundlage der Leistungsmetrik aufweist, die der Datenverarbeitungsressource zugehörig ist.

7. Computerrealisiertes Verfahren nach Anspruch 1, vor einem Starten des ersten vorläufigen PEs des Weiteren aufweisend: Starten eines zweiten vorläufigen PEs in einem dritten Knoten während der Laufzeit, während eine Laufzeitoperation des PEs beibehalten wird, wobei das zweite vorläufige PE den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfängt und den definierten Prozess für den ersten Datenstrom von Tupeln durchführt.

8. Computerrealisiertes Verfahren nach Anspruch 7, vor einem Starten des ersten vorläufigen PEs des Weiteren aufweisend: Vergleichen einer Operation des PEs mit einer Operation des zweiten vorläufigen PEs auf Grundlage

der Leistungsmetrik, die der Datenverarbeitungsressource zugehörig ist.

9. Computerrealisiertes Verfahren nach Anspruch 8, wobei das Vergleichen der Operation des PEs mit der Operation des zweiten vorläufigen PEs dazu führt, dass auf Grundlage der Leistungsmetrik identifiziert wird, dass die Operation des PEs die Operation des zweiten vorläufigen PEs übersteigt.

10. Computerprogrammprodukt aufweisend ein oder mehrere computerlesbare Speichermedien sowie Programmanweisungen, die gemeinsam auf dem einen oder den mehreren computerlesbaren Speichermedien gespeichert werden, wobei die Programmanweisungen durch einen Prozessor ausführbar sind, um den Prozessor zum Durchführen von Operationen zu veranlassen, aufweisend: Überwachen einer verteilten Datenverarbeitungsanwendung während der Laufzeit auf ein Eintreten einer Bedingung; wobei die verteilte Datenverarbeitungsanwendung einen ersten Knoten aufweist, der ein PE enthält, das einen ersten Datenstrom von Tupeln von einem vorgelagerten PE empfängt, einen definierten Prozess für den ersten Datenstrom von Tupeln durchführt, der zu einem zweiten Datenstrom von Tupeln führt, und den zweiten Datenstrom von Tupeln an ein nachgelagertes PE überträgt, und wobei die Bedingung ein Eintreten einer Leistungsabfallbedingung für das PE enthält; Starten eines ersten vorläufigen PEs in einem zweiten Knoten während der Laufzeit, während eine Laufzeitoperation des PEs beibehalten wird, wobei das erste vorläufige PE den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfängt und den definierten Prozess für den ersten Datenstrom von Tupeln durchführt; Vergleichen einer Operation des PEs mit einer Operation des ersten vorläufigen PEs auf Grundlage einer Leistungsmetrik, die einer Datenverarbeitungsressource zugehörig ist; und Verbinden des ersten vorläufigen PEs, um den zweiten Datenstrom von Tupeln von dem PE zu dem nachgelagerten PE durch einen dritten Datenstrom von Tupeln von dem ersten vorläufigen PE zu dem nachgelagerten PE zu ersetzen.

11. Computerprogrammprodukt nach Anspruch 10, wobei die gespeicherten Programmanweisungen in einer computerlesbaren Speichereinheit in einem Datenverarbeitungssystem gespeichert werden und wobei die gespeicherten Programmanweisungen über ein Netzwerk von einem entfernten angeordneten Datenverarbeitungssystem übertragen werden.

12. Computerprogrammprodukt nach Anspruch 10, wobei die gespeicherten Programmanweisungen

gen in einer computerlesbaren Speichereinheit in einem Server-Datenverarbeitungssystem gespeichert werden und wobei die gespeicherten Programmanweisungen als Reaktion auf eine Anforderung über ein Netzwerk in ein entfernt angeordnetes Datenverarbeitungssystem heruntergeladen werden, um in einer computerlesbaren Speichereinheit verwendet zu werden, die dem entfernt angeordneten Datenverarbeitungssystem zugehörig ist, des Weiteren aufweisend:

Programmanweisungen, um eine Nutzung der Programmanweisungen zu messen, die der Anforderung zugehörig sind; und

Programmanweisungen, um auf Grundlage der gemessenen Nutzung eine Rechnung zu erzeugen.

13. Computerprogrammprodukt nach Anspruch 10, des Weiteren aufweisend ein Auswählen des zweiten Knotens aus einer Mehrzahl von Knoten zum Ausführen des ersten vorläufigen PEs.

14. Computerprogrammprodukt nach Anspruch 10, wobei das erste vorläufige PE den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfängt und den definierten Prozess für den ersten Datenstrom von Tupeln durchführt, während das PE den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfängt und den definierten Prozess für den ersten Datenstrom von Tupeln durchführt.

15. Computerprogrammprodukt nach Anspruch 14, des Weiteren aufweisend ein Starten eines zweiten vorläufigen PEs in einem dritten Knoten während der Laufzeit, während eine Laufzeitoperation des PEs beibehalten wird, wobei das zweite vorläufige PE den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfängt und den definierten Prozess für den ersten Datenstrom von Tupeln durchführt, während das erste vorläufige PE und das PE jeweils den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfangen und jeweils den definierten Prozess für den ersten Datenstrom von Tupeln durchführen.

16. Computerprogrammprodukt nach Anspruch 15, wobei das Vergleichen der Operation des PEs mit der Operation des ersten vorläufigen PEs des Weiteren ein Vergleichen der Operation des zweiten vorläufigen PEs mit der Operation des PEs und mit der Operation des ersten vorläufigen PEs auf Grundlage der Leistungsmetrik aufweist, die der Datenverarbeitungsressource zugehörig ist.

17. Computersystem aufweisend einen Prozessor und ein oder mehrere computerlesbare Speichermedien sowie Programmanweisungen, die gemeinsam auf dem einen oder den mehreren computerlesbaren Speichermedien gespeichert werden, wobei die Programmanweisungen durch den Pro-

zessor ausführbar sind, um den Prozessor zum Durchführen von Operationen zu veranlassen, aufweisend:

Überwachen einer verteilten Datenverarbeitungsanwendung während der Laufzeit auf ein Eintreten einer Bedingung;

wobei die verteilte Datenverarbeitungsanwendung einen ersten Knoten aufweist, der ein PE enthält, das einen ersten Datenstrom von Tupeln von einem vorgelagerten PE empfängt, einen definierten Prozess für den ersten Datenstrom von Tupeln durchführt, der zu einem zweiten Datenstrom von Tupeln führt, und den zweiten Datenstrom von Tupeln an ein nachgelagertes PE überträgt, und wobei die Bedingung ein Eintreten einer Leistungsabfallbedingung für das PE enthält;

Starten eines ersten vorläufigen PEs in einem zweiten Knoten während der Laufzeit, während eine Laufzeitoperation des PEs beibehalten wird, wobei das erste vorläufige PE den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfängt und den definierten Prozess für den ersten Datenstrom von Tupeln durchführt;

Vergleichen einer Operation des PEs mit einer Operation des ersten vorläufigen PEs auf Grundlage einer Leistungsmetrik, die einer Datenverarbeitungsressource zugehörig ist; und

Verbinden des ersten vorläufigen PEs, um den zweiten Datenstrom von Tupeln von dem PE zu dem nachgelagerten PE durch einen dritten Datenstrom von Tupeln von dem ersten vorläufigen PE zu dem nachgelagerten PE zu ersetzen.

18. Computersystem nach Anspruch 17, des Weiteren aufweisend ein Auswählen des zweiten Knotens aus einer Mehrzahl von Knoten zum Ausführen des ersten vorläufigen PEs.

19. Computersystem nach Anspruch 17, wobei das erste vorläufige PE den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfängt und den definierten Prozess für den ersten Datenstrom von Tupeln durchführt, während das PE den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfängt und den definierten Prozess für den ersten Datenstrom von Tupeln durchführt.

20. Computersystem nach Anspruch 19, des Weiteren aufweisend ein Starten eines zweiten vorläufigen PEs in einem dritten Knoten während der Laufzeit, während eine Laufzeitoperation des PEs beibehalten wird,

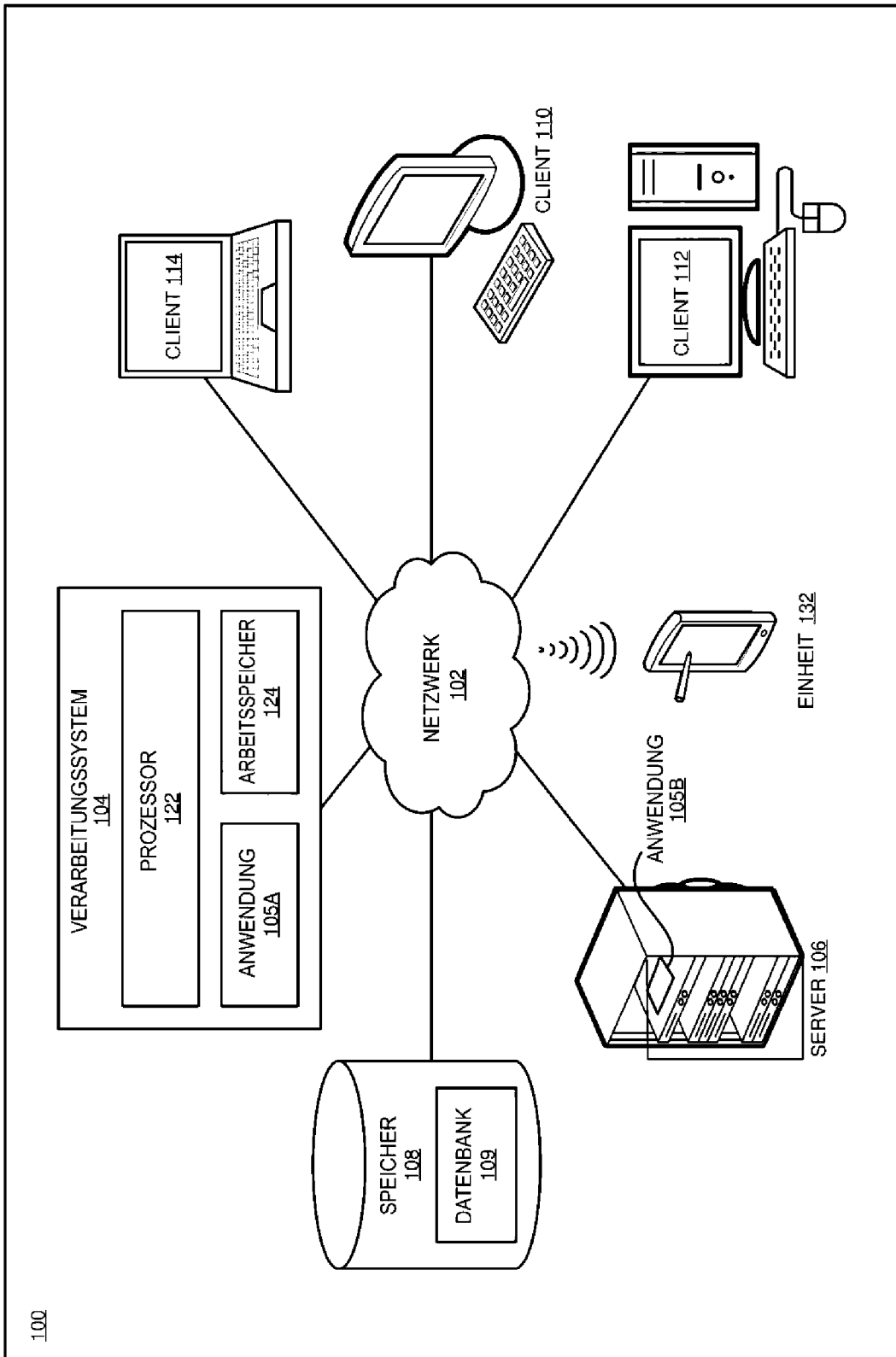
wobei das zweite vorläufige PE den ersten Datenstrom von Tupeln von dem vorgelagerten PE empfängt und den definierten Prozess für den ersten Datenstrom von Tupeln durchführt, während das erste vorläufige PE und das PE jeweils den ersten Datenstrom von Tupeln von dem vorgelagerten PE

empfangen und jeweils den definierten Prozess für den ersten Datenstrom von Tupeln durchführen.

Es folgen 12 Seiten Zeichnungen

Anhängende Zeichnungen

FIG. 1



100

FIG. 2

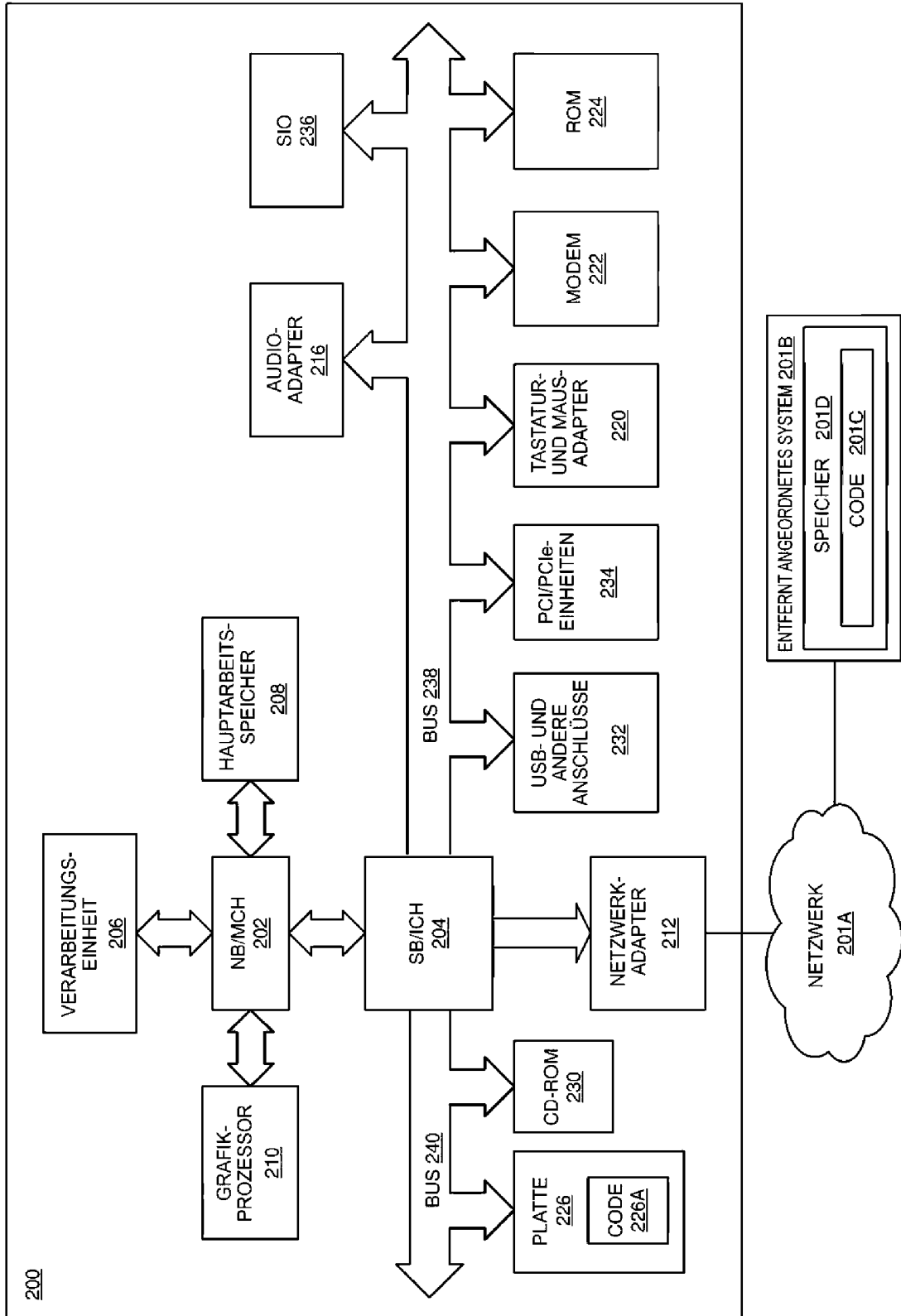


FIG. 3

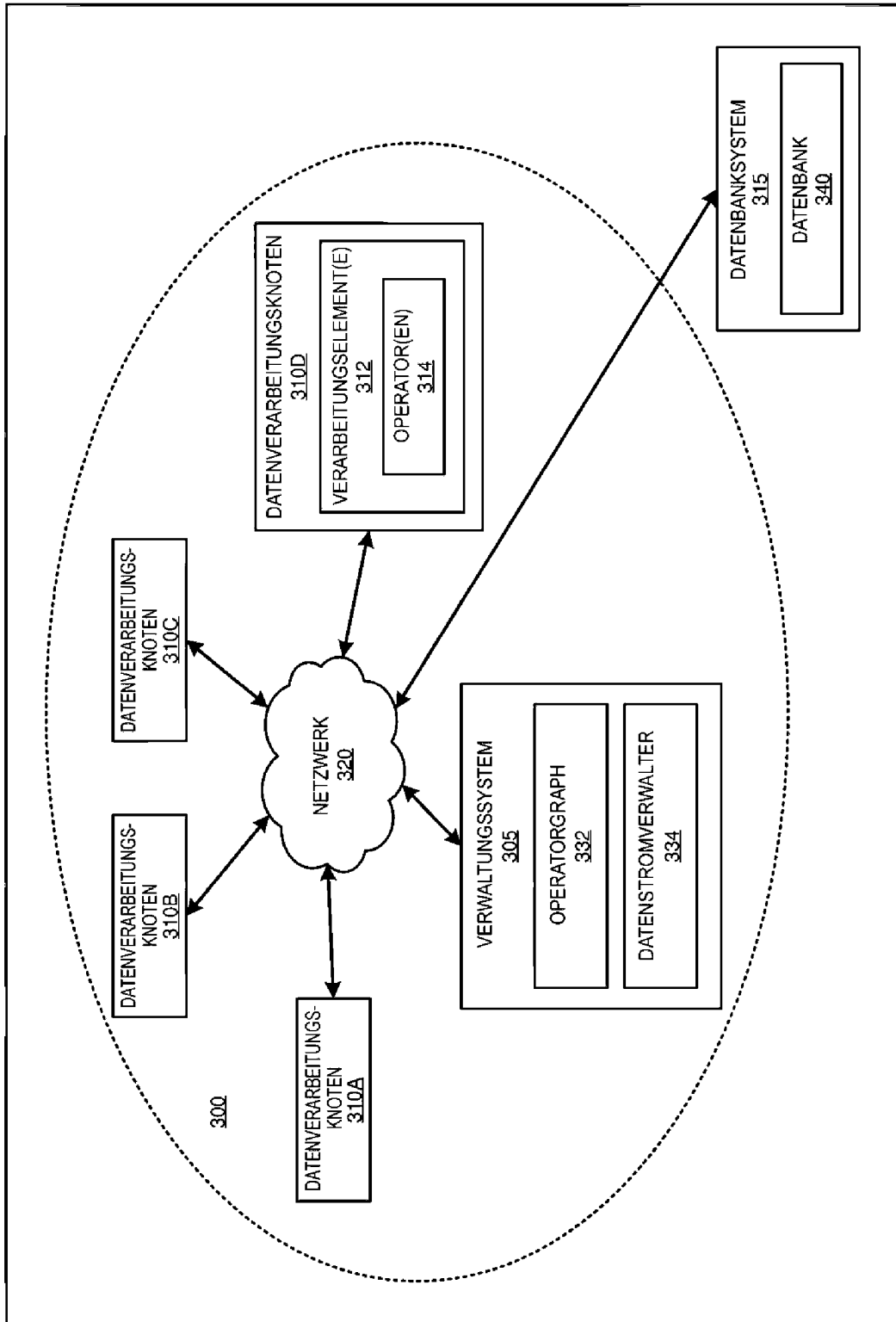


FIG. 4

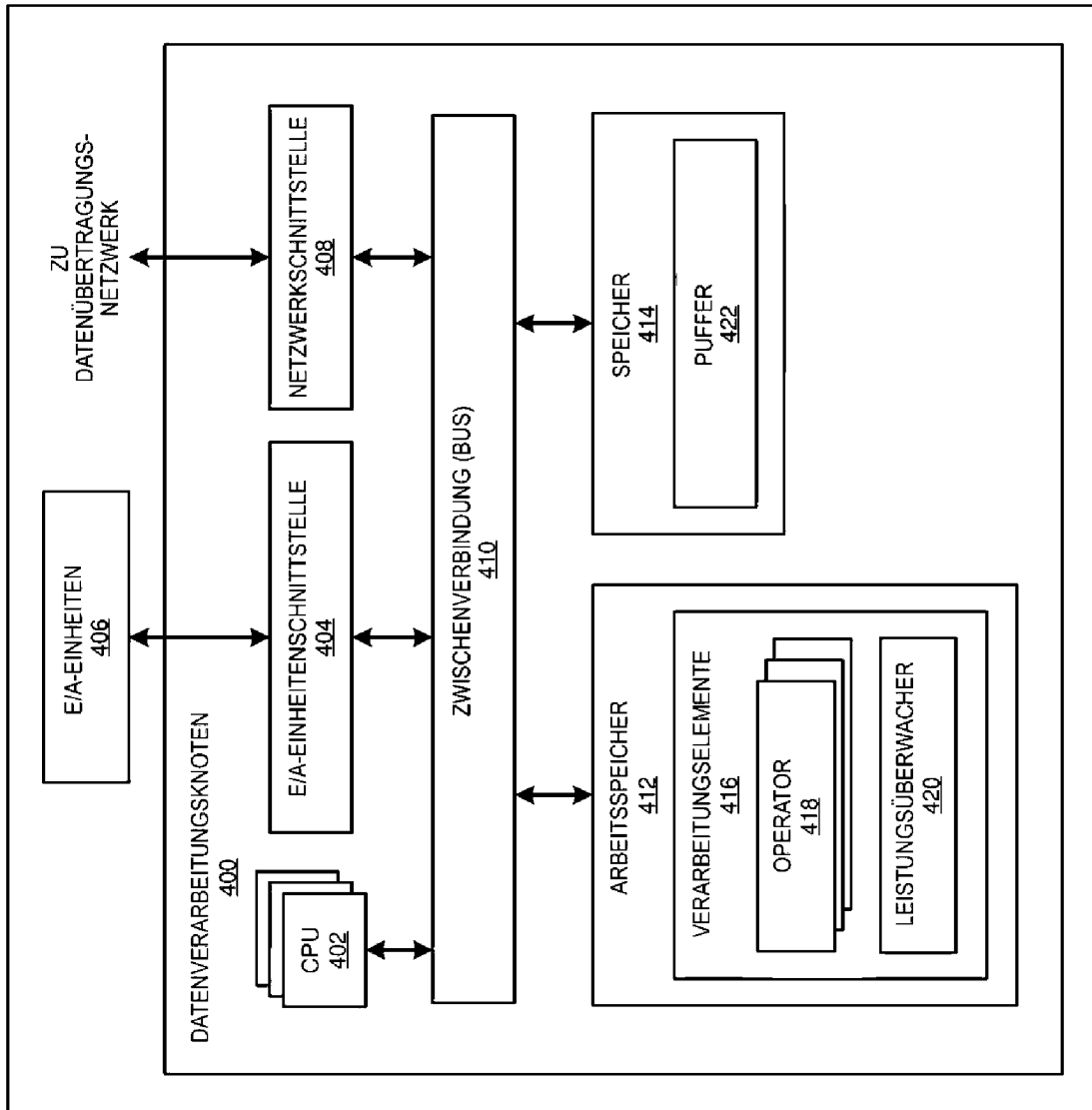


FIG. 5

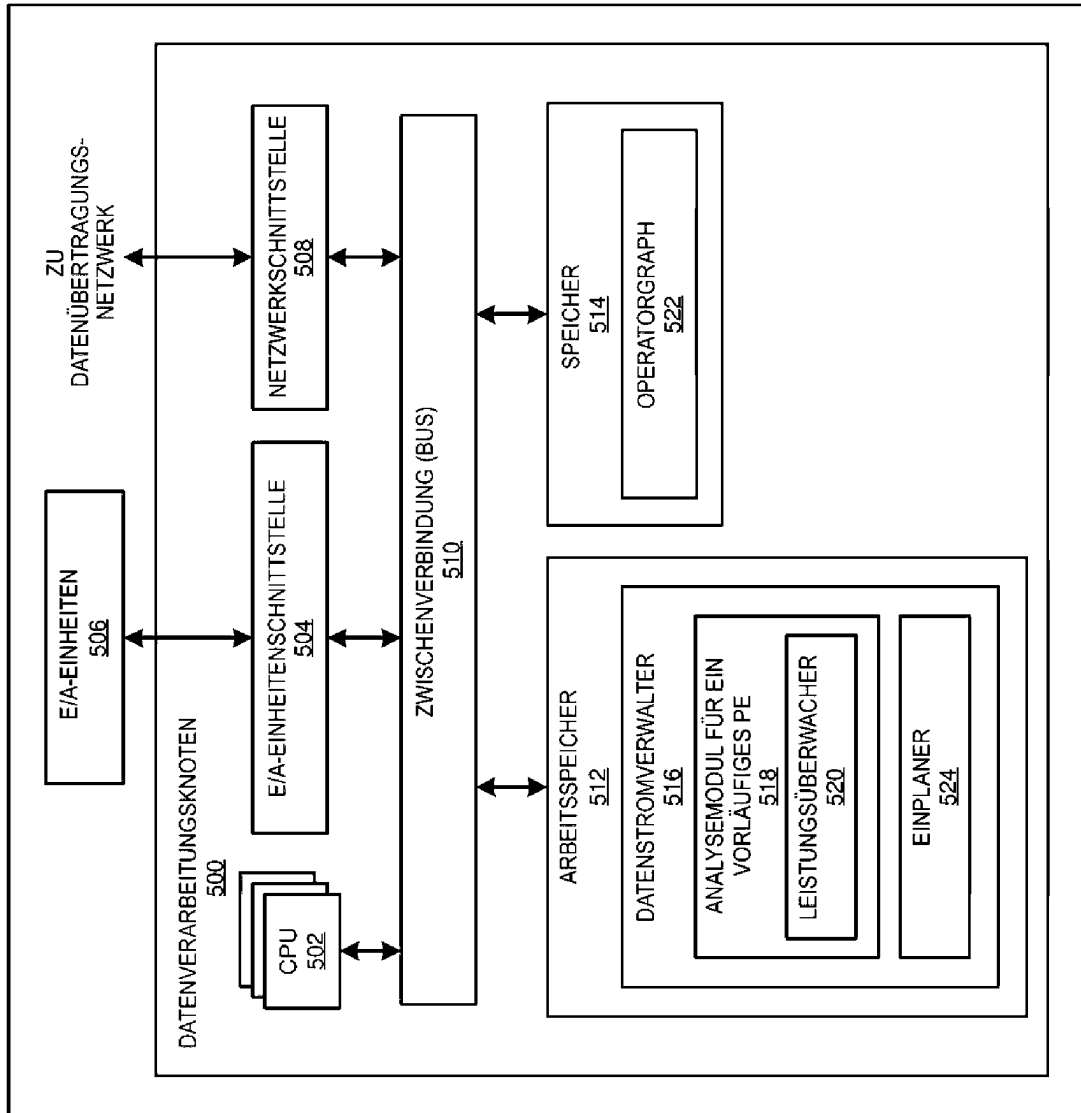


FIG. 6

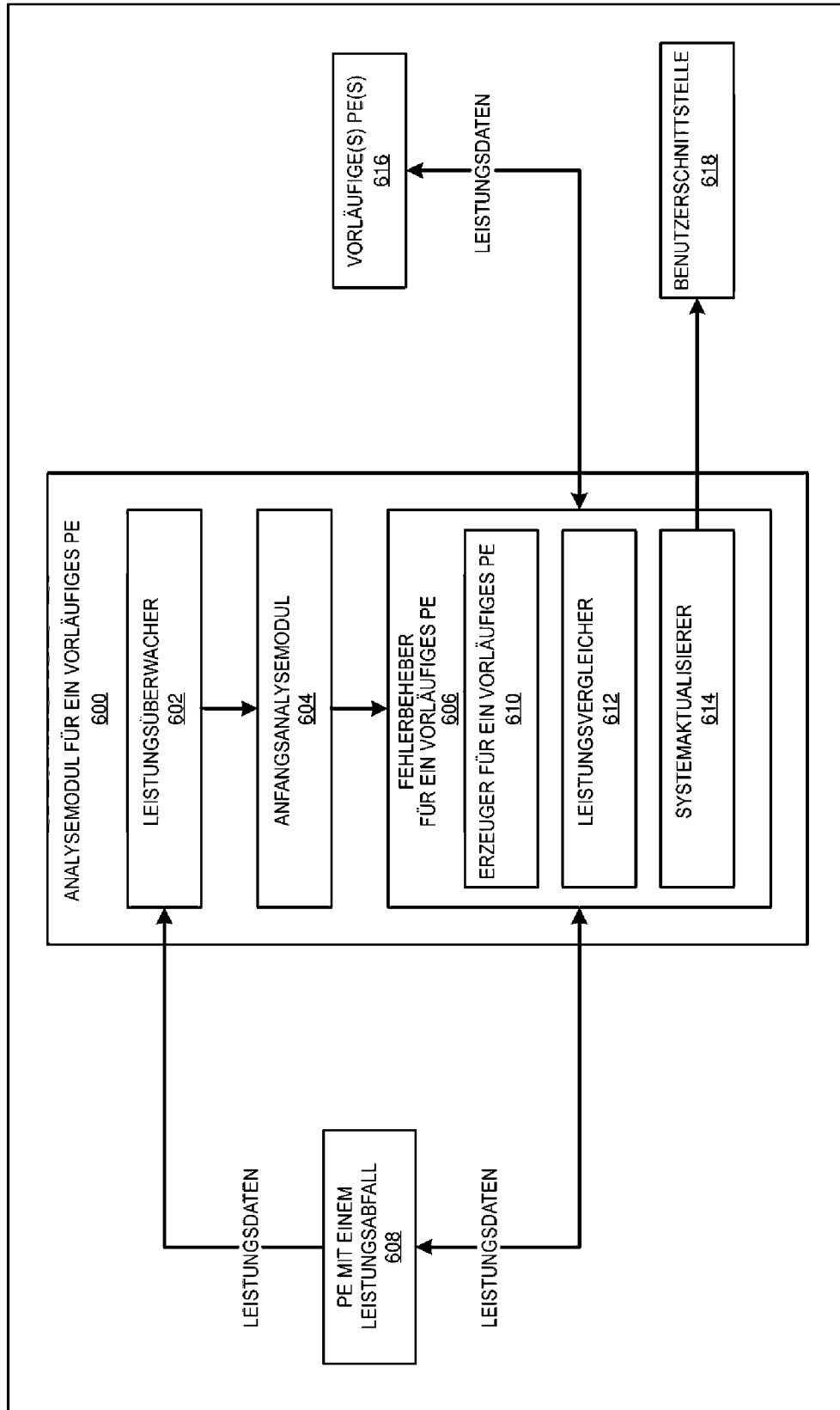


FIG. 7

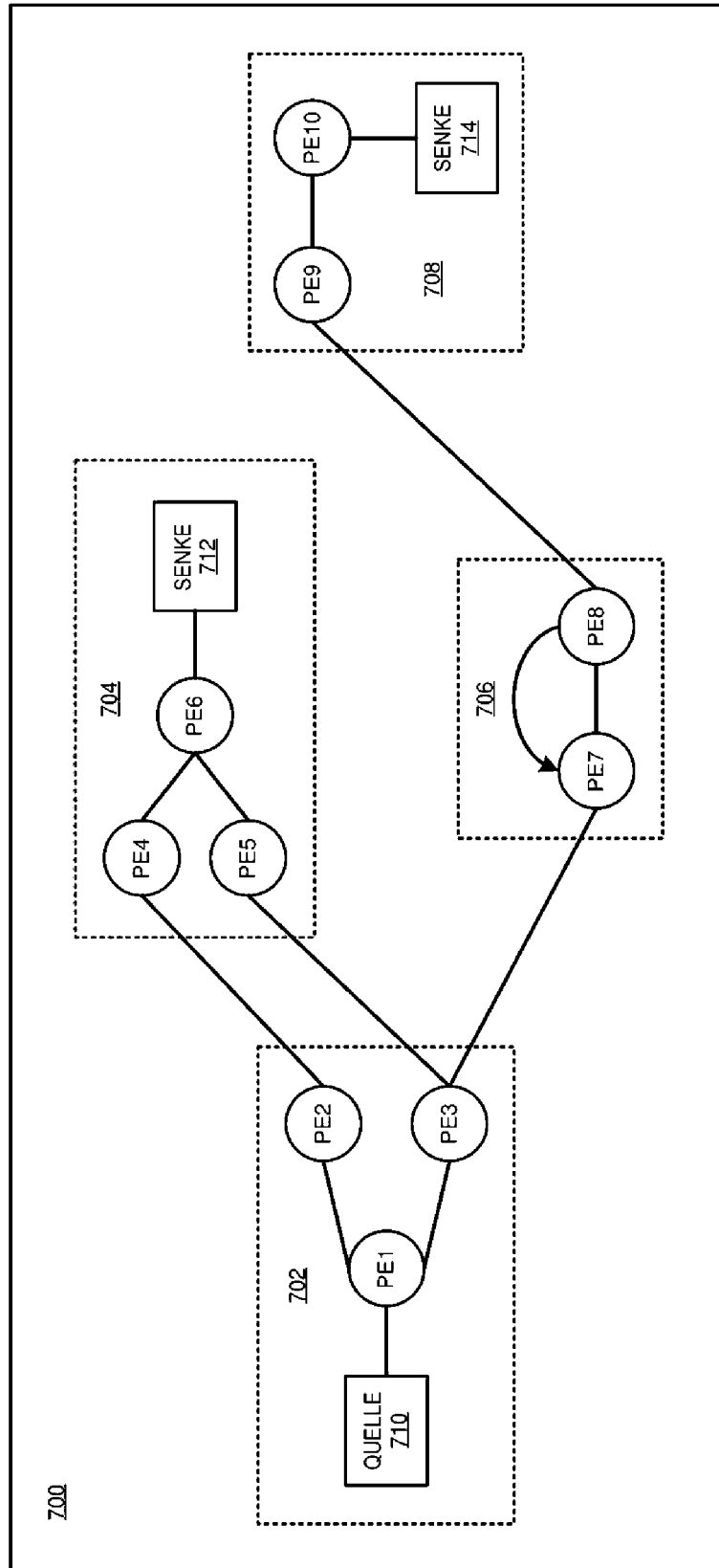


FIG. 8

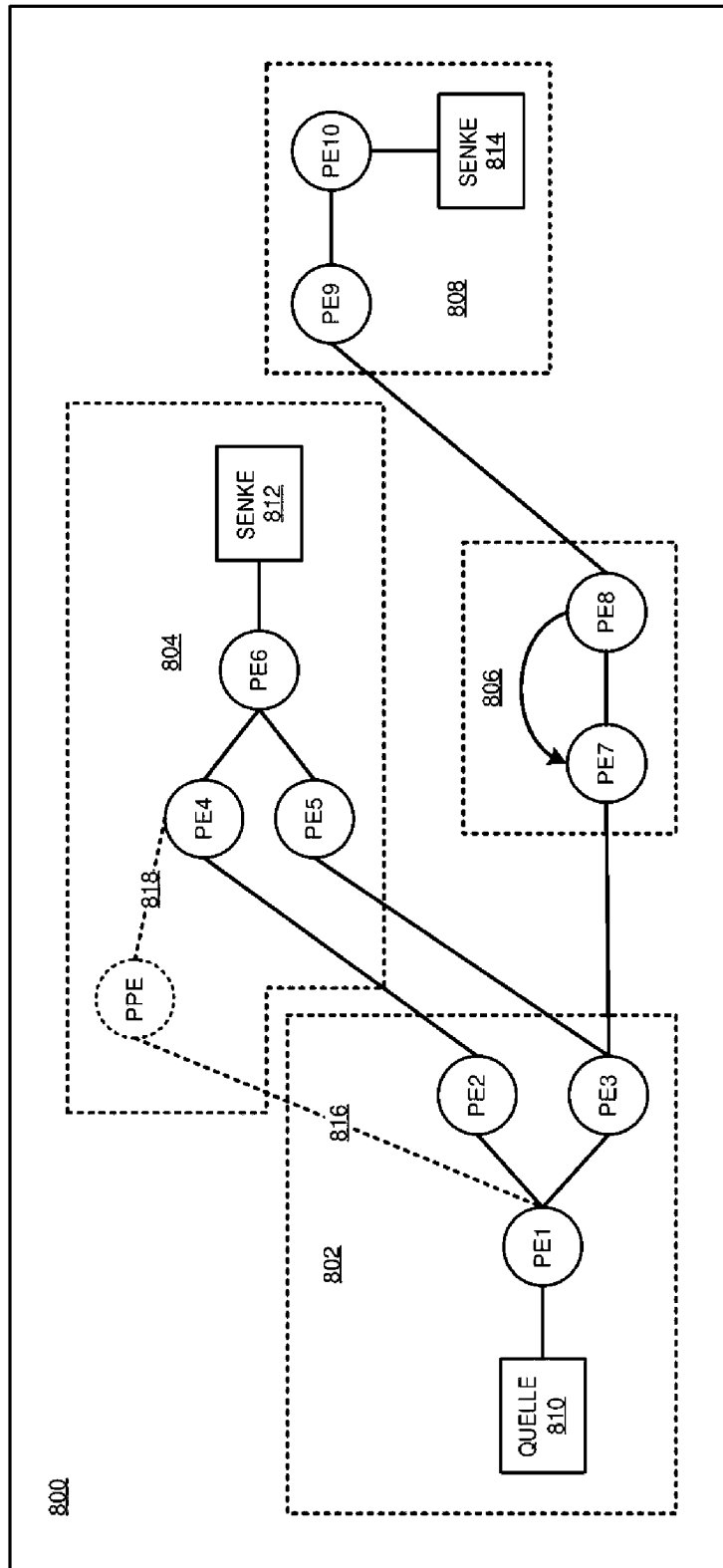


FIG. 9

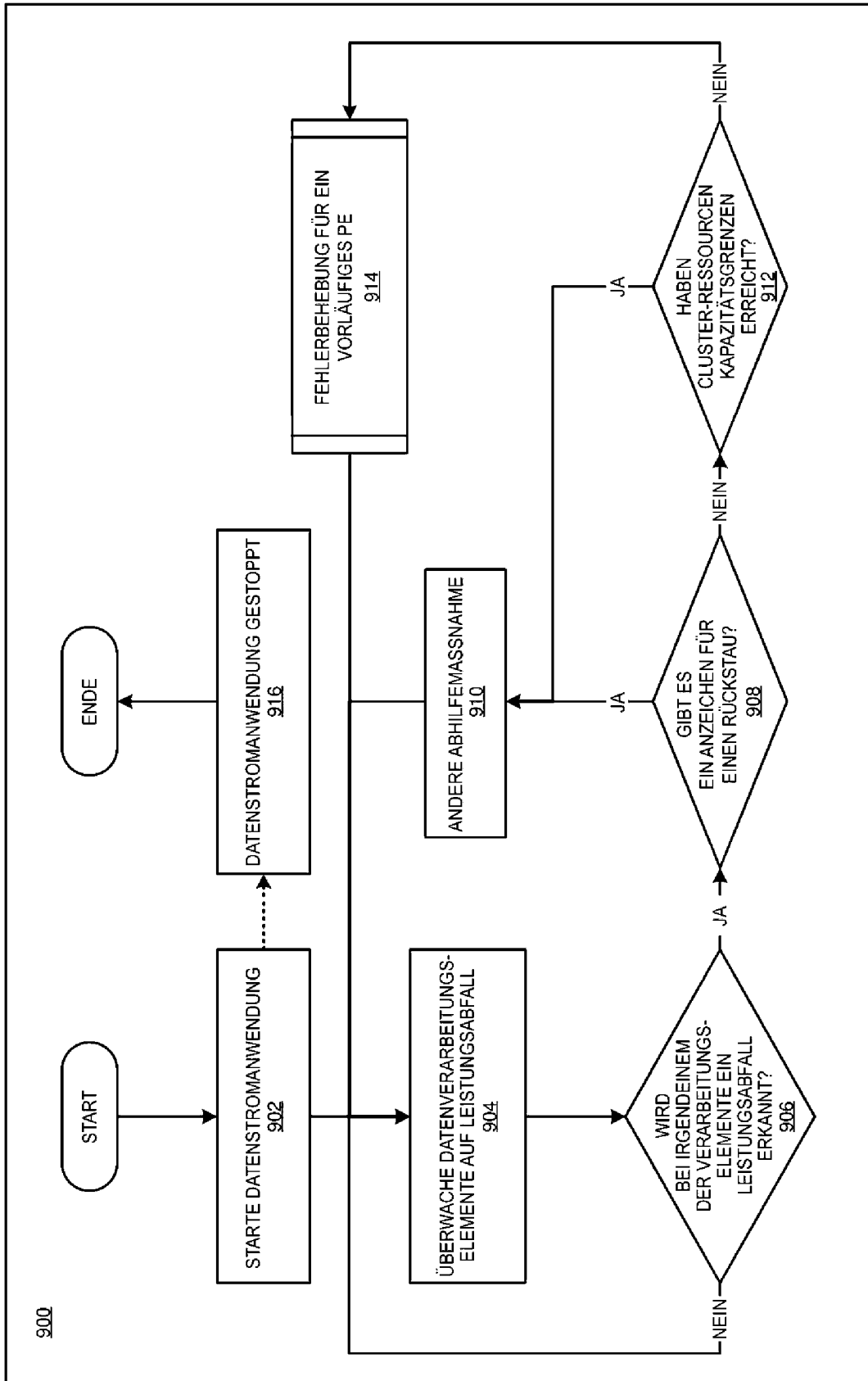


FIG. 10

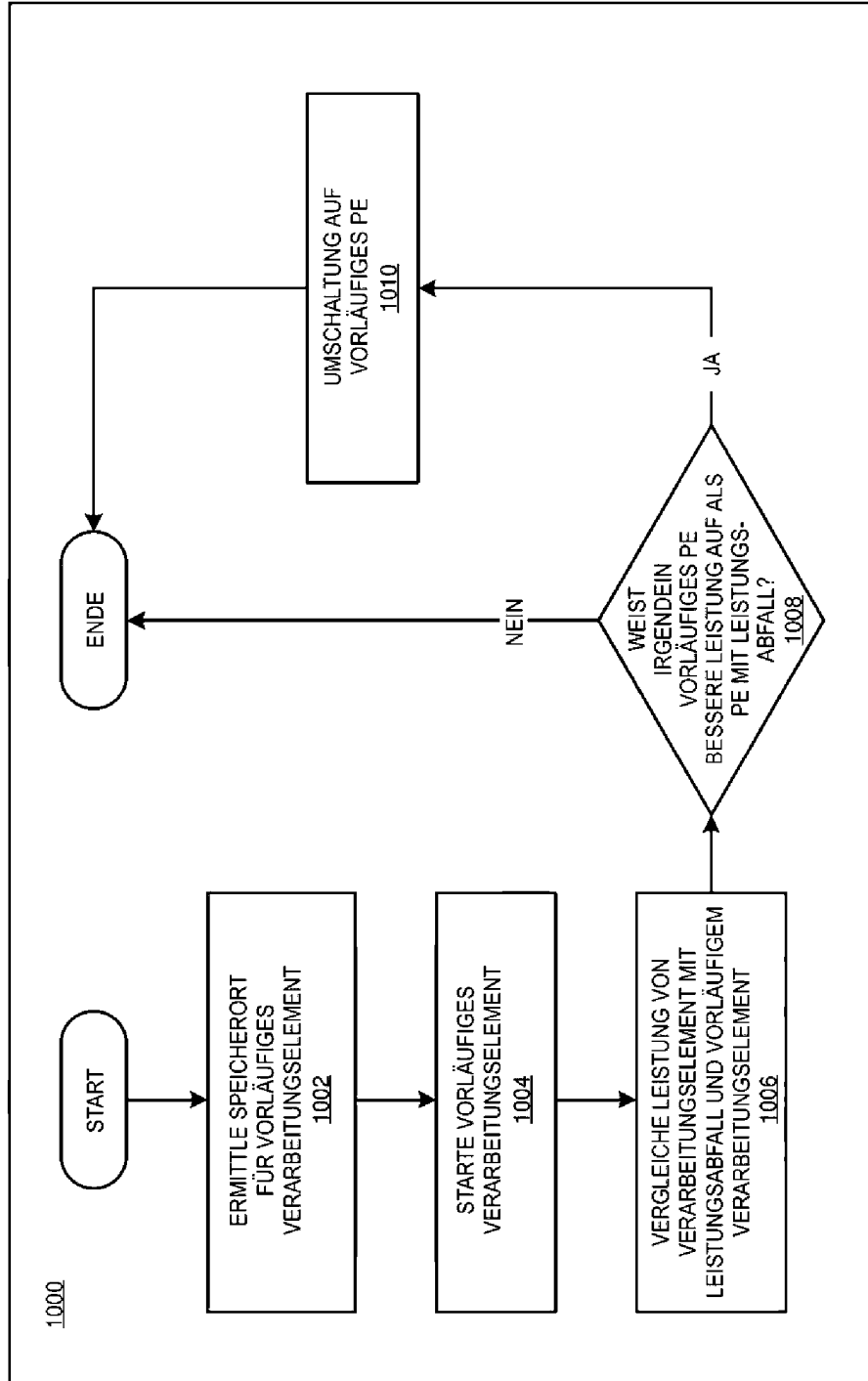


FIG. 11

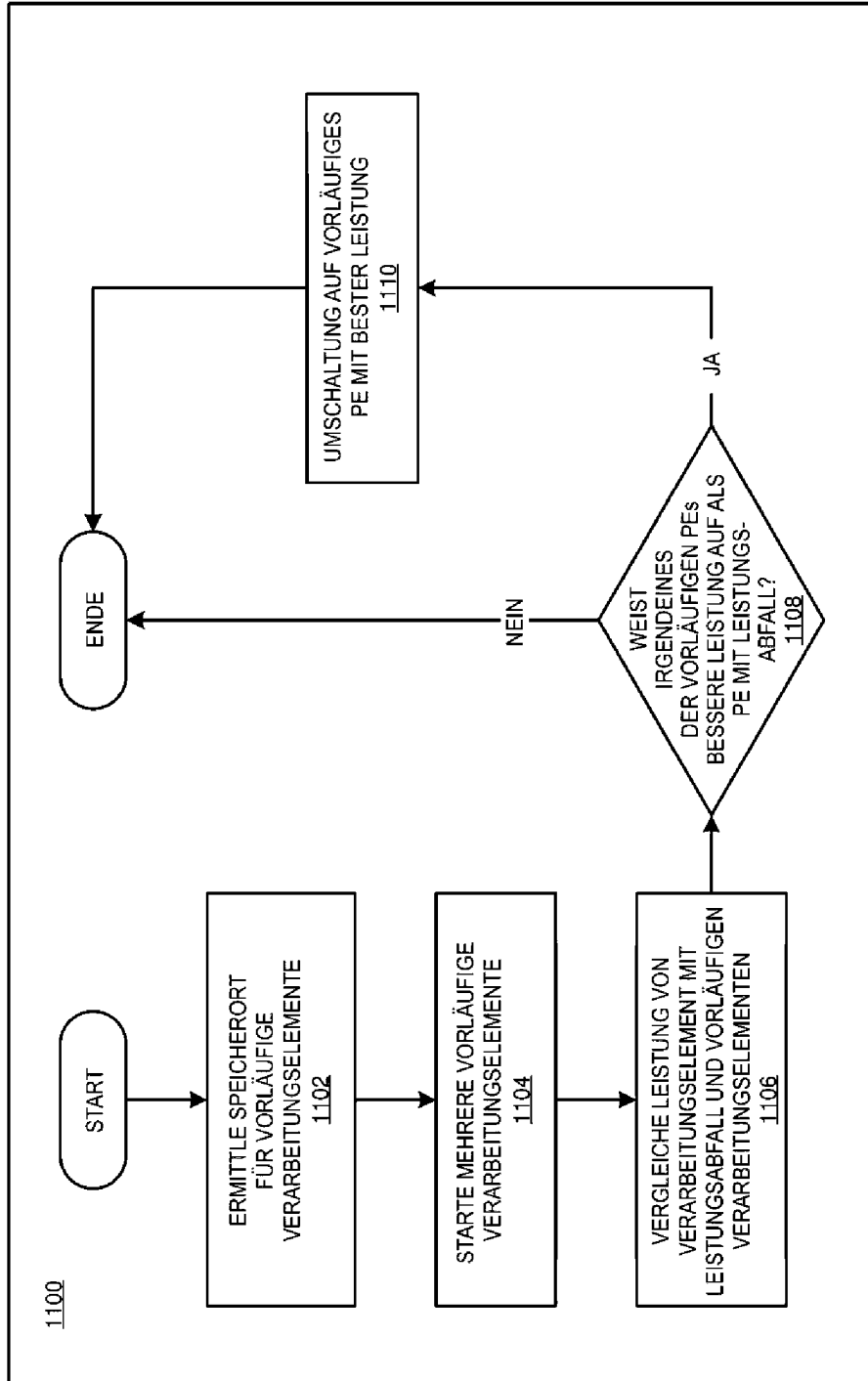


FIG. 12

