



(10) **DE 11 2014 005 348 T5** 2016.08.11

(12)

Veröffentlichung

der internationalen Anmeldung mit der
(87) Veröffentlichungs-Nr.: **WO 2015/122007**
in deutscher Übersetzung (Art. III § 8 Abs. 2 IntPatÜG)
(21) Deutsches Aktenzeichen: **11 2014 005 348.1**
(86) PCT-Aktenzeichen: **PCT/JP2014/053613**
(86) PCT-Anmeldetag: **17.02.2014**
(87) PCT-Veröffentlichungstag: **20.08.2015**
(43) Veröffentlichungstag der PCT Anmeldung
in deutscher Übersetzung: **11.08.2016**

(51) Int Cl.: **G06F 11/16 (2006.01)**
G06F 9/46 (2006.01)

(71) Anmelder:
Hitachi, Ltd., Tokyo, JP

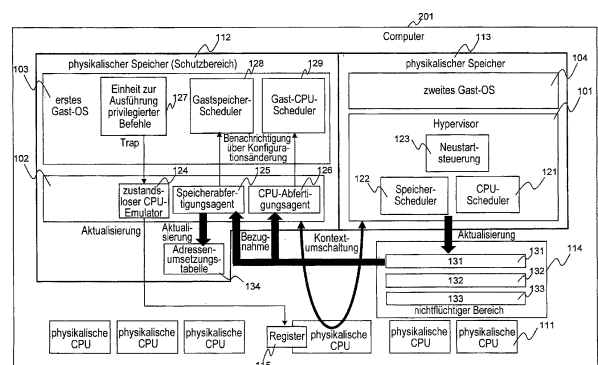
(74) Vertreter:
MERH-IP Matias Erny Reichl Hoffmann
Patentanwälte PartG mbB, 80336 München, DE

(72) Erfinder:
Tajima, Sachie, Tokyo, JP; Takeuchi, Tadashi,
Tokyo, JP

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Computer und Hypervisor-basiertes Betriebsmittelplanungsverfahren**

(57) Zusammenfassung: In einem Computer wird außer einem Hypervisor ein einfacher Hypervisor betrieben. In dem einfachen Hypervisor wird ein Gast-OS betrieben, dessen Weiterbetrieb garantiert werden muss, wenn in dem Hypervisor ein Fehler auftritt, und die anderen Gast-OS werden in dem Hypervisor betrieben. Der Hypervisor führt eine Betriebsmittelplanung (eine Bestimmung von Betriebsmitteln, die jedem Gast-OS zuzuweisen sind oder deren Zuweisung zu ihm aufzuheben ist) aus und der einfache Hypervisor führt anstelle des einfachen Hypervisors eine Zuweisung oder Aufhebung der Zuweisung von Betriebsmitteln zu oder von dem Gast-OS, deren Weiterbetrieb garantiert werden muss, aus.



Beschreibung

[Technisches Gebiet]

[0001] Die vorliegende Erfindung bezieht sich allgemein auf die Hypervisor-basierte Betriebsmittelplanung.

[Stand der Technik]

[0002] Ein Computer, der unter Verwendung eines Hypervisors mehrere Gast-OS betreibt, ist bekannt. In einem solchen Computer wird die Menge an Betriebsmitteln, die jedem Gast-OS zugewiesen werden, unter Verwendung der dynamischen Betriebsmittelplanungsfunktion des Hypervisors in Übereinstimmung mit einer Last eingestellt. Allgemeine Beispiele für Betriebsmittel enthalten physikalische Speicher und physikalische CPUs (Zentraleinheiten). Darüber hinaus ist ein Computer bekannt, in dem mehrere Gast-OS in demselben Gehäuse vorhanden sind.

[0003] In Abhängigkeit von einer Anwendung, auf die ein solcher Computer angewendet wird, ist eine garantierte Hochverfügbarkeit wesentlich. Ein Beispiel einer solchen Anwendung ist eine geschäftskritische Anwendung. Ein Hypervisor ist ein Mechanismus (üblicherweise ein Computerprogramm oder eine Hardwareschaltung, in die ein Computerprogramm eingebettet ist) mit vielen Funktionen. Somit ist es schwierig, die Verfügbarkeit des Hypervisors auf ein solches Niveau zu heben, dass der Hypervisor auf eine geschäftskritische Anwendung angewendet werden kann. Somit ist es allgemein schwierig, einen Hypervisor in einem geschäftskritischen Computer zu installieren.

[0004] NPL 1 offenbart ein Hypervisor-basiertes Betriebsmittelplanungsverfahren. In Übereinstimmung mit diesem Verfahren führt ein CPU-Scheduler (ein Beispiel für einen Betriebsmittel-Scheduler) eines Hypervisors dieselbe Operation wie ein CPU-Scheduler eines OS (Betriebssystems) aus.

[0005] Ein Hypervisor stellt virtuelle Ausführungsumgebungen (virtuelle Computer) bereit und behandelt Prozesse eines Gast-OS, das in einer virtuellen CPU jedes virtuellen Computers arbeitet, als Prozessoren. Der CPU-Scheduler in einer physikalischen CPU bestimmt, welches Gast-OS (welcher Prozessor) geplant werden soll. Ferner führt der Hypervisor außerdem einen Prozess (einen Betriebsmittel-Abfertigungs-/Vorbelegungsprozess) des Umschaltens des Ausführungskontexts in der physikalischen CPU aus.

[Liste der Entgegenhaltungen]

[Nicht-Patent-Literatur]

[0006]

[NPL 1] VMware Inc., "The CPU scheduler in VMware vSphere 5.1", VMware technical white paper, 2013.

[Zusammenfassung der Erfindung]

[Technisches Problem]

[0007] Wenn in dem Verfahren in dem Hypervisor ein Fehler auftritt, können alle Gast-OS nicht weiterarbeiten. Dies liegt daran, dass der Prozess des Abfertigungs und Vorbelegens von Betriebsmitteln (des Zuweisens und Aufhebens der Zuweisung von Betriebsmitteln) zu und von dem Gast-OS wegen eines Fehlers in dem Hypervisor nicht ausgeführt wird. Alle Gast-OS können keine Betriebsmittel (CPUs, Speicher und dergleichen) verwenden, die für Operationen erforderlich sind, und können nicht weiterarbeiten.

[Lösung des Problems]

[0008] In einem Computer arbeitet zusätzlich zu einem Hypervisor ein einfacher Hypervisor. Der Hypervisor weist eine Funktion zum Emulieren mehrerer Betriebsmittel einschließlich eines oder mehrerer physikalischer Speicher und mehrerer physikalischer CPUs und eine Funktion zum Planen von Betriebsmitteln (zum Bestimmen eines Betriebsmittels, das jedem Gast-OS zuzuweisen ist oder dessen Zuweisung von ihm aufzuheben ist) auf. Der einfache Hypervisor weist eine Agentenfunktion zum Zuweisen oder Aufheben der Zuweisung von Betriebsmitteln zu dem Gast-OS oder von dem Gast-OS anstelle des Hypervisors auf. In dem einfachen Hypervisor arbeitet ein erstes Gast-OS (z. B. ein Gast-OS, dessen weitere Operationen garantiert werden müssen, wenn in dem Hypervisor ein Fehler auftritt) und in dem Hypervisor arbeitet ein zweites Gast-OS (ein von dem ersten Gast-OS verschiedenes Gast-OS). Der Hypervisor führt die Betriebsmittelplanung aus und der einfache Hypervisor führt die Zuweisung oder Aufhebung der Zuweisung von Betriebsmitteln zu oder von dem ersten Gast-OS anstelle des einfachen Hypervisors aus.

[Vorteilhafte Wirkungen der Erfindung]

[0009] Die weiteren Operationen des ersten Gast-OS können garantiert werden, selbst wenn in dem Hypervisor ein Fehler auftritt.

[Kurzbeschreibung der Zeichnungen]

[0010] Fig. 1 stellt eine Systemkonfiguration dar, die in einer Ausführungsform der vorliegenden Erfindung genutzt wird.

[0011] Fig. 2 stellt eine Hardwarekonfiguration dar, die in einem praktischen Beispiel der vorliegenden Erfindung genutzt wird.

[0012] Fig. 3 stellt eine Datenstruktur eines Abfertigungs-/Vorbelegungsanforderungsprotokolls dar.

[0013] Fig. 4 stellt eine Datenstruktur eines CPU-Zuweisungszustands dar.

[0014] Fig. 5 stellt eine Datenstruktur eines Speicherzuweisungszustands dar.

[0015] Fig. 6 stellt eine Datenstruktur eines Hypervisor-Kontextablagebereichs dar.

[0016] Fig. 7 stellt eine Datenstruktur eines Hypervisoragent-Kontextablagebereichs dar.

[0017] Fig. 8 stellt eine Anordnung von Codebereichen und Stapelbereichen dar, die während eines Abfertigungs-/Vorbelegungsprozesses verwendet werden.

[0018] Fig. 9 stellt eine Datenstruktur einer Adressenumsetzungstabelle dar.

[0019] Fig. 10 stellt eine Datenstruktur einer CPU-Nutzungshistorien-Managementtabelle dar.

[0020] Fig. 11 stellt eine Datenstruktur einer Speichernutzungshistorien-Managementtabelle (1001) dar.

[0021] Fig. 12 stellt einen Ablaufplan eines CPU-Schedulers (während eines regulären Starts) dar.

[0022] Fig. 13 stellt einen Betriebsablauf eines CPU-Schedulers dar (wenn von einem CPU-Abfertigungsagenten ein Trap-Rücksprungbefehl eingereicht wird).

[0023] Fig. 14 stellt einen Betriebsablauf eines CPU-Abfertigungsagenten und des CPU-Schedulers eines ersten Gast-OS dar (wenn eine Abfertigungsanforderung eingereicht wird).

[0024] Fig. 15 stellt einen Betriebsablauf eines CPU-Abfertigungsagenten und des CPU-Schedulers eines ersten Gast-OS dar (wenn eine Vorbelegungsanforderung eingereicht wird).

[0025] Fig. 16 stellt einen Betriebsablauf eines Speicher-Schedulers dar.

[0026] Fig. 17 stellt einen Betriebsablauf eines Speicherabfertigungsagenten und des Speicher-Schedulers eines ersten Gast-OS dar.

[0027] Fig. 18 stellt einen Betriebsablauf einer Einheit zur Ausführung privilegierter Befehle und eines zustandslosen CPU-Emulators dar.

[0028] Fig. 19 stellt einen Betriebsablauf einer Neustartsteuerung (während des regulären Starts) dar.

[0029] Fig. 20 stellt einen Betriebsablauf einer Neustartsteuerung (während des Neustarts) dar.

[Beschreibung von Ausführungsformen]

[0030] Im Folgenden wird eine Ausführungsform beschrieben.

[0031] Obwohl Informationen in der folgenden Beschreibung häufig als der Ausdruck einer "kkk-Tabelle" beschrieben sind, können Informationen als eine andere Datenstruktur als die Tabelle ausgedrückt werden. Um zu zeigen, dass die Informationen nicht von einer Datenstruktur abhängen, kann die "kkk-Tabelle" als "kkk-Informationen" bezeichnet werden.

[0032] In der folgenden Beschreibung kann es Fälle geben, dass die Verarbeitung unter Verwendung eines "Programms" als das Objekt beschrieben ist. Da die Verarbeitung allerdings ausgeführt wird, während ein Ablagebetriebsmittel (z. B. ein Speicher) und/oder eine Kommunikationsschnittstellenvorrichtung (z. B. ein Kommunikationsport) verwendet werden, wie sie notwendig sind, wenn ein Programm durch eine physikalische CPU ausgeführt wird, kann die physikalische CPU ebenfalls als das Objekt der Verarbeitung verwendet werden. Umgekehrt ist zu verstehen, dass die unter Verwendung der physikalischen CPU als das Objekt beschriebene Verarbeitung durch Ausführung eines oder mehrerer Programme ausgeführt wird. Darüber hinaus kann eine physikalische CPU eine Hardwareschaltung enthalten, die einen Teil oder alle der Prozesse ausführt, die durch den Prozessor ausgeführt werden, und die jeder Kern eines Mehrkernprozessors sein kann. Ein Computerprogramm kann von einer Programmquelle in jeweiligen Computer installiert werden. Die Programmquelle kann z. B. ein Programmverteilungsserver oder ein Ablagemedium sein.

[0033] Fig. 1 stellt eine Übersicht über eine Ausführungsform dar.

[0034] Ein Computer 201 enthält mehrere physikalische CPUs 111 und mehrere Betriebsmittel (physikalische Betriebsmittel), einschließlich physikalischer Speicher 112 und 113. Der physikalische Speicher 112 ist ein Beispiel eines ersten physikalischen Speicherbereichs und der physikalische Speicher 113 ist ein Beispiel eines zweiten physikalischen Speicherbereichs. Die physikalischen Speicher 112 und 113 können mehrere Speicherbereiche sein, die von

Speicherbereichen eines integrierten physikalischen Speichers gesichert sind.

[0035] Zusätzlich zu einem Hypervisor **101** enthält der Computer **201** einen Hypervisoragenten **102**, wobei sie eine virtuelle Computerumgebung bereitstellen. Der Hypervisoragent **102** ist ein Beispiel eines einfachen Hypervisors. In einen virtuellen Computer, der durch den Hypervisoragenten **102** bereitgestellt wird, arbeitet ein erstes Gast-OS **103** und in einem virtuellen Computer, der durch den Hypervisor **101** bereitgestellt wird, arbeitet ein zweites Gast-OS **104**. Die Betriebsmittelplanung der dynamischen Zuweisung oder Aufhebung der Zuweisung von Betriebsmitteln wird in dem ersten und in dem zweiten Gast-OS **103** und **104** ausgeführt. In der vorliegenden Ausführungsform enthalten Beispiele für Betriebsmittel, die dynamisch zugewiesen werden oder deren Zuweisung dynamisch aufgehoben wird, physikalische CPUs und Speicherbereiche (in der vorliegenden Ausführungsform gelegentlich als "Seiten" bezeichnet).

[0036] Das für Operationen des Hypervisoragenten **102** verwendete Betriebsmittel kann unter den mehreren Betriebsmitteln nur das Betriebsmittel sein, das das erste Gast-OS **103** verwenden kann (d. h., das der Hypervisor **101** nicht verwendet). Der Hypervisoragent **102** kann durch Aktualisieren einer Adressenumsetzungstabelle **134**, die die Entsprechung zwischen einer logischen Adresse (in der vorliegenden Ausführungsform der Nummer einer physikalischen Seite des Gasts), die von einem Gast-OS bestimmt wird, und einer physikalischen Adresse (in der vorliegenden Ausführungsform einer Nummer der physikalischen Seite des Hosts), die der logischen Adresse entspricht, zeigt, virtualisieren. Auf diese Weise ist es möglich, einen Speicherbereich (eine Seite), auf die das erste Gast-OS **103** zugreifen kann, zu steuern. Die Adressenumsetzungstabelle **134** kann z. B. eine EPT-Tabelle (Erweiterungsseitentabelle) oder eine DWAR-Tabelle (DMA-Neuzuordnungstabelle (Speicherdirektzugriffs-Neuzuordnungstabelle)) sein.

[0037] Obwohl der Hypervisor **101** die Betriebsmittelplanung bestimmt, wird die Zuweisung oder die Aufhebung der Zuweisung von Betriebsmitteln in Bezug auf das erste Gast-OS **103** anstatt durch den Hypervisor durch den Hypervisoragenten **102** ausgeführt.

[0038] Der Hypervisor **101** reicht durch Schreiben der Anforderung in einen nichtflüchtigen Bereich (nichtflüchtigen Ablagebereich) **114** eine Betriebsmittel-Abfertigungs-/Vorbelegungsanforderung (Zuweisungs-/Zuweisungsaufhebungsanforderung) ein. Der Hypervisoragent **102** liest die Anforderung aus dem nichtflüchtigen Bereich **114** und führt einen der Anforderung entsprechenden Prozess aus. Anstelle des

nichtflüchtigen Bereichs **114** kann ein flüchtiger Ablagebereich genutzt werden. Darüber hinaus kann der nichtflüchtige Bereich **114** außerhalb des Computers **201** (siehe Fig. 2) vorgesehen sein.

[0039] Falls die Anforderung eine CPU-Abfertigungsanforderung ist, wird der Kontext der physikalischen CPU eines Abfertigungsziels von dem Hypervisor **101** zu dem Hypervisoragenten **102** umgeschaltet und arbeitet die physikalische CPU in dem Hypervisoragenten **102**. Ferner benachrichtigt der Hypervisoragent **102** das erste Gast-OS **103**, dass die physikalische CPU verwendbar geworden ist. Im Ergebnis kann das erste Gast-OS **103** in der physikalischen CPU zu arbeiten beginnen.

[0040] Falls die Anforderung eine Vorbelegungsanforderung der physikalischen CPU ist, benachrichtigt der Hypervisoragent **102** das erste Gast-OS **103**, dass die physikalische CPU unverwendbar geworden ist. Im Ergebnis arbeitet das erste Gast-OS **103** in der physikalischen CPU nicht. Ferner wird der Kontext der physikalischen CPU des Vorbelegungsziels von dem Hypervisoragenten **102** zu dem Hypervisor **101** umgeschaltet.

[0041] Falls die Anforderung eine Speicherabfertigungsanforderung oder eine Speichervorbelegungsanforderung ist, wird ein Prozess des Aktualisierens der Adressenumsetzungstabelle **134** in der Weise, dass der Zugriff des ersten Gast-OS **103** auf den bestimmten Speicherbereich freigegeben oder gesperrt wird, ausgeführt. In diesem Fall sendet der Hypervisoragent **102** an das erste Gast-OS **103** eine Benachrichtigung und kann das erste Gast-OS **103** erkennen, dass der Zugriff auf den Speicherbereich freigegeben oder gesperrt ist.

[0042] In der durch den Hypervisoragenten **102** bereitgestellten virtuellen Computerumgebung werden die physikalischen Speicher **112** und **113** virtualisiert, während die physikalischen CPUs nicht virtualisiert werden. Das heißt, das erste Gast-OS **103** belegt ein Register **115** in der physikalischen CPU **111** und führt es aus, während es das Register allein aktualisiert. Allerdings greift das erste Gast-OS **103** mit Hilfe der Adressenumsetzungstabelle **134** auf die physikalischen Speicher **112** und **113** zu. In diesem Fall wird der Adressenraum des physikalischen Speichers umgesetzt und kann das erste Gast-OS **103** die Anwesenheit eines teilweisen physikalischen Speichers (Speicherbereichs) nicht erkennen.

[0043] Um eine dynamische Betriebsmittelplanung zu verwirklichen, enthält der Hypervisor **101** einen CPU-Scheduler **121** und einen Speicher-Scheduler **122**. Diese Scheduler bestimmen die Zuweisung der physikalischen CPU **111** und der physikalischen Speicher **112** und **113** zu dem ersten und zu dem zweiten Gast-OS **103** und **104**.

[0044] Der Hypervisoragent **102** enthält einen CPU-Abfertigungsagenten **126**, einen Speicherabfertigungsagenten **125** und einen zustandslosen CPU-Emulator **124**.

[0045] Der Scheduler führt den Prozess des Abfertigungs der physikalischen CPU (**111**) und der physikalischen Speicher **112** und **113** für das erste Gast-OS nicht allein aus, sondern reicht mit Hilfe des nichtflüchtigen Bereichs **114** eine Betriebsmittel-Abfertigungs-/Vorbelegungsanforderung ein. Durch Bezugnahme auf diese Anforderung führt der CPU-Abfertigungsagent **126** oder der Speicherabfertigungsagent **125** des Hypervisoragenten **102** den CPU/Speicher-Abfertigungsprozess aus.

[0046] Ein CPU-Abfertigungsprozess umfasst das Umschalten des Ausführungskontexts der physikalischen CPU **111**. Auf diese Weise wird das in der physikalischen CPU **111** arbeitende Gast-OS umgeschaltet. Darüber hinaus umfasst ein Speicherabfertigungsprozess das Einstellen (Aktualisieren) der Adressenumsetzungstabelle **134**. Auf diese Weise ist es möglich, neue Bereiche des physikalischen Speichers **112**, auf die von dem ersten Gast-OS **103** zugegriffen werden soll, teilweise freizugeben oder zu sperren. Das erste Gast-OS **103** enthält einen Gast-CPU-Scheduler **129**, einen Gast-Speicher-Scheduler **128** und eine Einheit **127** zur Ausführung privilegierter Befehle. Um den Abfertigungsprozess zu verwirklichen, führt der Hypervisoragent **102** eine Zusammenarbeit mit dem Gast-CPU-Scheduler **129** und mit dem Speicher-Scheduler **128** aus.

[0047] Mit der oben beschriebenen Konfiguration ist es möglich, eine dynamische Planung von CPU/Speicher-Betriebsmitteln zu verwirklichen, die von dem ersten und von dem zweiten Gast-OS **103** und **104** verwendet werden.

[0048] In einer in der vorliegenden Ausführungsform genutzten Systemkonfiguration führt das erste Gast-OS **103** Prozesse direkt in der physikalischen CPU **111** aus. Allerdings können privilegierte Befehle des ersten Gast-OS **103** in Abhängigkeit von der physikalischen CPU **111** abgefangen werden, wenn eine Speichervirtualisierungsfunktion der Adressenumsetzungstabelle **134** verwendet wird. Um in solchen Fällen die direkte Ausführung zu garantieren, enthält der Hypervisoragent **102** den oben beschriebenen zustandslosen CPU-Emulator **124**. Der zustandslose CPU-Emulator **124** führt einen Prozess des Aktualisierens des Registers **115** in der physikalischen CPU **111** in Abhängigkeit von dem abgefangenen privilegierten Befehl aus.

[0049] Um zu ermöglichen, dass das erste Gast-OS **103** und der Hypervisoragent **102** weiterarbeiten, und um zu ermöglichen, dass der Hypervisor **101** und das zweite Gast-OS **104** neu starten und Prozesse wei-

ter ausführen, selbst wenn in dem Hypervisor **101** ein Fehler auftritt, sind der Hypervisoragent **102** und das erste Gast-OS **103** in einem Schutzbereich des physikalischen Speichers **112** (d. h. in einem Bereich, in den der Hypervisor **101** und das zweite Gast-OS **104** keine Daten schreiben können) angeordnet. In diesem Bereich können der Hypervisor **101** und das zweite Gast-OS **104** keine Daten oder Codes zerstören. Darüber hinaus enthält der Hypervisor **101** eine Neustartsteuerung **123**. Der nichtflüchtige Bereich **114** speichert eine Anforderungsprotokolltabelle **131**, eine CPU-Zuweisungszustandstabelle **132** und eine Speicher-Zuweisungszustandstabelle **133** und die Neustartsteuerung **131** nimmt auf diese in dem nichtflüchtigen Bereich **114** gespeicherten Informationseinheiten **131** bis **133** Bezug. Die Neustartsteuerung **123** führt eine Bereitschaft aus, bis die den in der Anforderungsprotokolltabelle **131** registrierten eingereichten Anforderungen entsprechenden Prozesse abgeschlossen sind. Ferner führt die Neustartsteuerung **123** diese Startprozesse nur unter Verwendung der CPUs/Speicher aus, die der Hypervisor **101** und das zweite Gast-OS **104** in Übereinstimmung mit einem CPU-/Speicherzuweisungszustand verwenden können.

[0050] Die Abfertigung von Betriebsmitteln, die durch das erste Gast-OS **103** verwendet werden, wird anstatt durch den Hypervisor **101** durch den Hypervisoragenten **102** ausgeführt. Darüber hinaus wird die CPU-Emulation während der Ausführung des ersten Gast-OS **103** anstatt durch den Hypervisor **101** ebenfalls durch den Hypervisoragenten **102** ausgeführt. Das heißt, selbst wenn in dem Hypervisor **101** ein Fehler auftritt, können das erste Gast-OS **103** und der Hypervisoragent **102** weiterarbeiten. Die durch den Hypervisoragenten **102** bereitgestellten Funktionen sind kleiner (z. B. weitaus kleiner) als jene des Hypervisors **101**. Somit wird betrachtet, dass das erste Gast-OS **103** leicht weiterarbeiten kann, selbst wenn in dem Hypervisor **101** ein Fehler auftritt.

[0051] Die Anforderungsprotokolltabelle **131**, die CPU-Zuweisungszustandstabelle **132** und die Speicher-Zuweisungszustandstabelle **133** sind in dem nichtflüchtigen Bereich **114** gespeichert, so dass auf diese Tabellen sogar Bezug genommen werden kann, nachdem der Hypervisor **101** neu gestartet worden ist. Mit dieser Konfiguration können das zweite Gast-OS **104** und der Hypervisor **101** sogar erst neu starten und die Ausführung von Prozessen fortsetzen, nachdem in dem Hypervisor ein Fehler aufgetreten ist.

[0052] Im Folgenden wird die vorliegende Ausführungsform ausführlich beschrieben.

[0053] Fig. 2 stellt eine Hardwarekonfiguration des Computers **201** dar.

[0054] Programme wie etwa das erste Gast-OS **103**, der Hypervisoragent **102**, das zweite Gast-OS **104** und der Hypervisor **101** sind in den physikalischen Speichern **112** und **113** des Computers **201** angeordnet. Die physikalischen Speicher **112** und **113** sind über einen CPU-Bus **202** mit mehreren physikalischen CPUs **111** verbunden, um miteinander zu kommunizieren. Wenigstens eine der mehreren physikalischen CPUs **111** liest die Programme und führt sie aus. Die Adressenumsetzungstabelle **134** ist in den physikalischen Speichern **112** und **113** angeordnet und wird zum Steuern der Adressenumsetzung für den Speicherzugriff verwendet, wenn das erste und das zweite Gast-OS **103** und **104** ausgeführt werden. Die Anforderungsprotokolltabelle **131**, die CPU-Zuweisungszustandstabelle **132**, die Speicher-Zuweisungszustandstabelle **133** sind in dem nichtflüchtigen Bereich **114** angeordnet. Der Computer **102** enthält einen E/A-Controller (Eingabe/Ausgabe-Controller) **203**, wobei von einem in der physikalischen CPU **111** ausgeführten Programm über den E/A-Controller **203** und ein E/A-Kabel **204** auf den nichtflüchtigen Bereich **114** zugegriffen werden kann. Zum Beispiel reicht das Programm eine E/A-Anforderung an den E/A-Controller **203** ein, wenn Leseoperationen ausgeführt werden. Der E/A-Controller **203** liest über das E/A-Kabel **204** Daten in dem nichtflüchtigen Bereich **114** und schreibt die gelesenen Daten über den CPU-Bus **202** in die physikalischen Speicher **112** und **113**. Ein in der physikalischen CPU **111** arbeitendes Programm erfasst die über den CPU-Bus **202** geschriebenen Daten. Das E/A-Kabel **204** kann ein Kabel eines Kommunikationsnetzes sein.

[0055] Fig. 3 stellt eine Konfiguration der Anforderungsprotokolltabelle **131** dar.

[0056] Die Anforderungsprotokolltabelle **131** ist eine Tabelle zum Managen von Protokollen einschließlich einer Betriebsmittel-Abfertigungs-/Vorbelegungsanforderung, die durch den CPU-Scheduler **121** und durch den Speicher-Scheduler **122** ausgegeben wird, und eines Verarbeitungszustands der Anforderung durch den CPU-Abfertigungsagenten **126** und durch den Speicheragenten **125**. Diese Tabelle enthält ein Betriebsmittelfeld **301**, ein Nummernfeld **302**, ein Feld **303** des alten Zustands, ein Feld **304** des neuen Zustands, ein Feld **305** der Nummer des physikalischen Gasts und ein Verarbeitungszustandsfeld **306** für jeweilige Anforderungen. Eine Spalte entspricht den Protokollen einer Anforderung.

[0057] Das Betriebsmittelfeld **301** gibt an, ob ein Abfertigungs-/Vorbelegungsziel eine CPU ist.

[0058] Das Nummernfeld **302** gibt eine Nummer (eine Kennnummer) eines Abfertigungs-/Vorbelegungsziel-Betriebsmittels an. Wenn eine CPU ein Abfertigungsziel ist, gibt das Nummernfeld **302** die Nummer der physikalischen CPU eines Abfertigungsziels an.

Wenn ein Speicher ein Abfertigungsziel ist, gibt das Nummernfeld **302** die Nummer der physikalischen Seite des Hosts eines Abfertigungsziel-Speicherbereichs (einer Abfertigungsziel-Speicherseite) und eine Seitenkennnummer an.

[0059] Das Feld **303** des alten Zustands und das Feld **304** des neuen Zustands geben die Nummern eines Gast-OS an, das das Zielbetriebsmittel vor bzw. nach dem Abfertigungs-/Vorbelegungsprozess verwenden kann. In einem leeren Zustand, in dem das Zielbetriebsmittel keinem Gast-OS zugewiesen ist, können das Feld **303** des alten Zustands und/oder das Feld **304** des neuen Zustands einen Wert "F" aufweisen, der den leeren Zustand angibt.

[0060] Das Feld **305** der physikalischen Nummer des Gasts wird nur in Spezialfällen verwendet, in denen der Hypervisor **101** anders als in dem Fall, dass der Hypervisor **101** eine Betriebsmittel-Abfertigungs-/Vorbelegungsanforderung einreicht, an den Hypervisoragenten **102** eine Anforderung zum Einstellen der Adressenumsetzungstabelle **134** einreicht. In der vorliegenden Ausführungsform wird die Adressenumsetzungstabelle **134** durch den Hypervisor **101** und durch den Hypervisoragenten **102** gemeinsam genutzt. In diesem Fall ist die Adressenumsetzungstabelle **134** in dem physikalischen Speicher (Schutzbereich) **112** angeordnet, um zu verhindern, dass die Adressenumsetzungstabelle **134** durch den Hypervisor **101** zerstört wird. Wenn der Hypervisor **101** die Adressenumsetzungstabelle **134** aktualisiert, reicht der Hypervisor **101** an den Speicherabfertigungsagenten **125** eine Adressenänderungsanforderung ein, in der Informationen (z. B. das Nummernfeld **302** und das Feld **304** des neuen Zustands), die für die Einstellung erforderlich sind, bestimmt sind.

[0061] Das Verarbeitungszustandsfeld **306** gibt einen Verarbeitungszustand des der Anforderung entsprechenden Abfertigungs-/Vorbelegungsprozesses an. Wenn eine Anforderung eingereicht wird, aktualisieren der CPU-Scheduler **121** und der Speicher-Scheduler **122** das der Anforderung entsprechende Verarbeitungszustandsfeld **306** in "in Verarbeitung". Darüber hinaus aktualisieren der CPU-Abfertigungsagent **126** und der Speicherabfertigungsagent **125** das der Anforderung entsprechende Verarbeitungszustandsfeld **306** auf "abgeschlossen", wenn der der Anforderung entsprechende Prozess abgeschlossen wird.

[0062] In Abhängigkeit von dem Anforderungstyp werden nur Abschnitte der Felder verwendet. Zum Beispiel wird das Feld **205** der physikalischen Nummer des Gasts nicht verwendet, wenn die Anforderung eine CPU-Abfertigungs-/Vorbelegungsanforderung ist.

[0063] Es werden drei Speicher-Abfertigungs-/Vorbelegungsanforderungen verwendet, die eine Abfertigungsanforderung zum Zuweisen eines Speicherbereichs (einer Speicherseite) zu dem ersten Gast-OS **103**, eine Vorbelegungsanforderung zum Aufheben der Zuweisung eines Speicherbereichs von dem Gast-OS **103** und eine Adressenaktualisierungsanforderung zum Aktualisieren der durch den Hypervisor **101** eingereichten Adressen der Adressenumsetzungstabelle **134** enthalten. Im Fall der Abfertigungsanforderung wird das Feld **305** der physikalischen Nummer des Gasts nicht verwendet. Im Fall der Vorbelegungsanforderung werden das Nummernfeld **302** und das Feld **305** der physikalischen Nummer des Gasts nicht verwendet (die Anforderung kann nur die Zuweisung einer optionalen Seite aufheben, die von dem ersten Gast-OS **103** aktuell verwendet ist). Darüber hinaus wird im Fall der Adressenumschaltanforderung das Feld **303** des alten Zustands nicht verwendet (das liegt daran, dass sich der Eigentümer des Betriebsmittels, bevor und nachdem die Anforderung eingereicht worden ist, nicht geändert hat).

[0064] Fig. 4 stellt eine Konfiguration der CPU-Zuweisungszustandstabelle **132** dar.

[0065] Die CPU-Zuweisungszustandstabelle **132** enthält ein CPU-Nummernfeld **401** und ein Zuweisungszielfeld **402** für jeweilige physikalische CPUs. Das CPU-Nummernfeld **401** gibt eine Nummer (Kennnummer) der physikalischen CPU **111** an und das Zuweisungszielfeld **402** gibt eine Nummer (Kennnummer) des Gast-OS, dem die entsprechende physikalische CPU **111** zugewiesen ist, an. Das Zuweisungszielfeld **402** der physikalischen CPU **111**, deren Zuweisungsziel-Gast-OS nicht vorhanden ist und das leer ist, kann einen Wert "F" aufweisen, der einen leeren Zustand angibt. Darüber hinaus kann das Zuweisungszielfeld **402** für die physikalische CPU **111**, die in dem Abfertigungs-/Vorbelegungsprozess ist, einen Wert "Nr." aufweisen, der angibt, dass die physikalische CPU **111** in Verarbeitung ist.

[0066] Fig. 5 stellt eine Konfiguration der Speicher-Zuweisungszustandstabelle **133** dar.

[0067] Die Speicher-Zuweisungszustandstabelle **133** enthält ein Feld **501** der Nummer der physikalischen Seite des Hosts, ein Feld **502** der Nummer der physikalischen Seite des Gasts und ein Zuweisungszielfeld **503** für jeweilige Seiten. Das Feld **501** der Nummer der physikalischen Seite des Hosts gibt eine Nummer (Kennnummer) einer Seite an und entspricht einer physikalischen Adresse. Das Feld **502** der Nummer der physikalischen Seite des Gasts gibt eine durch das Gast-OS erkannte Seitennummer an und entspricht einer logischen Adresse. Das Zuweisungszielfeld **503** gibt eine Nummer des Gast-OS an, dem die Seite zugewiesen ist. Für Seiten, deren Zuweisungsziel-Gast-OS nicht vorhanden ist und die

leer sind, können das Feld **502** der Nummer der physikalischen Seite des Gasts und das Zuweisungszielfeld **503** einen Wert "F" aufweisen, der einen leeren Zustand angibt. Darüber hinaus können das Feld **502** der Nummer der physikalischen Seite des Gasts und das Zuweisungszielfeld **503** für Seiten, die in dem Abfertigungs-/Vorbelegungsprozess sind, einen Wert "Nr." aufweisen, der angibt, dass die Seiten in Verarbeitung sind.

[0068] Fig. 6 stellt einen ersten Kontextablagebereich **601** dar.

[0069] Der erste Kontextablagebereich **601** kann ein Teilbereich des physikalischen Speichers **113** sein. Der erste Kontextablagebereich **601** wird verwendet, wenn der CPU-Scheduler **121** einen CPU-Abfertigungs-/Vorbelegungsprozess ausführt. Der CPU-Scheduler **121** sichert unter Verwendung des ersten Kontextablagebereichs **601** den Ausführungskontext des Hypervisoragenten **102** und gewinnt ihn wieder.

[0070] Fig. 7 stellt einen zweiten Kontextablagebereich **701** dar.

[0071] Der zweite Kontextablagebereich **701** kann ein Teilbereich des physikalischen Speichers **112** sein. Der CPU-Abfertigungsagent **126** sichert den Ausführungskontext des Hypervisors **101** in der Datenstruktur, bevor er eine CPU-Abfertigungsanforderung einreicht. Darüber hinaus wird der Ausführungskontext des Hypervisors **101** aus dem zweiten Kontextablagebereich **701** wiedergewonnen, nachdem der der Vorbelegungsanforderung entsprechende Prozess abgeschlossen ist.

[0072] Fig. 8 stellt die Anordnung von Codebereichen und Stapelbereichen dar.

[0073] Ein Codebereich und ein Stapelbereich werden verwendet, wenn ein Prozess des Abfertigungs und des Vorbelegens der physikalischen CPU **111** ausgeführt wird. Durch den CPU-Scheduler **121** wird eine CPU-Abfertigungs-/Vorbelegungsanforderung eingereicht. Ein Code zum Ausführen des Prozesses ist in einem CPU-Scheduler-Codebereich (1) **813** angeordnet. Darüber hinaus ist ein Stapel, der beim Ausführen des Prozesses verwendet wird, in einem CPU-Scheduler-Stapelbereich **815** angeordnet. Diese Bereiche sind in dem physikalischen Speicher **113** angeordnet.

[0074] Die Abfertigungsanforderung der physikalischen CPU **111** wird durch Einreichen eines Trap-Befehls eingereicht. In diesem Fall wird ein Sprungziel in einer Unterbrechungsbehandlertabelle **801** registriert, die durch den Hypervisor **101** gemanagt wird, und wird mit Hilfe der Tabelle **801** ein Sprung zu dem CPU-Abfertigungsagenten **126** verwirklicht. Codes eines durch den CPU-Abfertigungsagenten

126 ausgeführten Prozesses sind in einem CPU-Abfertigungsagent-Codebereich **812** angeordnet. In diesem Fall tritt das Umschalten von Stapeln gleichzeitig auf und werden neue Stapel in einem CPU-Abfertigungsagent-Stapelbereich **811** angeordnet. Diese Bereiche sind in dem physikalischen Speicher (Schutzbereich **112**) angeordnet.

[0075] Eine CPU-Vorbelegungsanforderung wird durch Senden einer Unterbrechungsbenachrichtigung an eine physikalische CPU **111** des Vorbelegungsziels eingereicht. Nachdem eine Unterbrechung in der physikalischen CPU **111** des Vorbelegungsziels aufgetreten ist, finden ähnlich wie oben die Bezugnahme, der Sprung und das Stapelumschalten der Unterbrechungsbehandlertabelle **801** statt und wird der CPU-Abfertigungsagent **126** gestartet.

[0076] Wenn der CPU-Abfertigungsagent **126** den Vorbelegungsprozess abschließt, kehrt die Steuerung durch den Trap-Rücksprung zu dem CPU-Scheduler **121** zurück. Die Codes der danach ausgeführten Prozesse sind in einem CPU-Scheduler-Codebereich (2) **814** angeordnet. In diesem Fall werden die zu verwendenden Stapel in die in einem CPU-Scheduler-Stapelbereich **815** vorhandenen Stapel geändert. Diese Bereiche sind in dem physikalischen Speicher **113** angeordnet.

[0077] Wenn der CPU-Abfertigungsagent **126** den Abfertigungsprozess abschließt, kehrt die Steuerung nicht zu dem CPU-Scheduler **121** zurück, da die abgefertigte physikalische CPU **111** belegt ist und durch den Hypervisoragenten **102** und durch das erste Gast-OS **103** ausgeführt wird. Wenn der Vorbelegungsprozess abgeschlossen ist, kehrt die Steuerung zu dem CPU-Scheduler **121** zurück.

[0078] Fig. 9 stellt eine Konfiguration der Adressenumsetzungstabelle **134** dar.

[0079] Die Adressenumsetzungstabelle **134** enthält die folgenden Felder für jeweilige Einträge (Datensätze), einschließlich eines V-Felds **901**, das einen Wert speichert, der angibt, ob ein Eintrag gültig oder ungültig ist, eines Felds **902** der Nummer des Gast-OS, das eine Nummer des Gast-OS eines Seitenzuweisungsziels angibt, eines Felds **903** der Nummer der physikalischen Seite des Gasts, das eine von einem Gast-OS bestimmte Seitennummer angibt, und eines Felds **904** der Nummer der physikalischen Seite des Hosts, das mit einer Nummer der physikalischen Seite des Gasts korreliert ist.

[0080] Fig. 10 stellt eine Konfiguration einer CPU-Nutzungshistorien-Managementtabelle **1001** dar.

[0081] Die CPU-Nutzungshistorien-Managementtabelle **1001** ist eine Tabelle, die durch den CPU-Sche-

duler **121** gemanagt wird, und ist z. B. in dem physikalischen Speicher **113** angeordnet. Die CPU-Nutzungshistorien-Managementtabelle **1001** enthält ein Feld **1011** der Nummer des Gast-OS, ein Feld **1012** der kleinsten CPU-Menge, ein Feld **1013** der größten CPU-Menge und ein Feld **1014** der zugewiesenen CPU-Gesamtheit für jeweilige Gast-OS. Das Feld **1011** der Nummer des Gast-OS gibt eine Nummer des Gast-OS an, das Feld **1012** der kleinsten CPU-Menge gibt eine Untergrenze der dem Gast-OS zugewiesenen CPU-Menge an, das Feld **1013** der größten CPU-Menge gibt eine Obergrenze der dem Gast-OS zugewiesenen CPU-Menge an und das Feld **1014** der zugewiesenen CPU-Gesamtheit gibt eine Gesamtheit der zugewiesenen CPU-Mengen an. Die CPU-Mengen können z. B. die Anzahlen physikalischer CPUs sein. Eine Kombination des Felds **1012** der kleinsten CPU-Menge und des Felds **1013** der größten CPU-Menge gibt einen Bereich der dem Gast-OS zugewiesenen Mengen der CPU an.

[0082] Fig. 11 stellt eine Konfiguration einer Speichernutzungshistorien-Managementtabelle **1101** dar.

[0083] Die Speichernutzungshistorien-Managementtabelle **1101** ist eine Tabelle, die durch den Speicher-Scheduler **122** gemanagt wird und die z. B. in dem physikalischen Speicher **113** angeordnet ist. Die Speichernutzungshistorien-Managementtabelle **1101** enthält ein Feld **1111** der Nummer des Gast-OS, ein Feld **1112** der kleinsten Speichermenge, ein Feld **1113** der größten Speichermenge und ein Feld **1114** der zugewiesenen Speichergesamtheit für jeweilige Gast-OS. Das Feld **1111** der Nummer des Gast-OS gibt eine Nummer des Gast-OS an, das Feld **1112** der kleinsten Speichermenge gibt eine Untergrenze einer dem Gast-OS zugewiesenen Speichermenge an, das Feld **1113** der größten Speichermenge gibt eine Obergrenze der dem Gast-OS zugewiesenen Speichermenge an und das Feld **1114** der zugewiesenen Speichergesamtheit gibt eine Gesamtheit der zugewiesenen Speichermengen an. Die Speichermenge kann z. B. die Anzahl der Seiten oder das Gesamtvolumen der Seiten sein. Eine Kombination der Werte des Felds **1112** der kleinsten Speichermenge und des Felds **1113** der größten Speichermenge gibt einen Bereich der dem Gast-OS zugewiesenen Speichermengen an.

[0084] Die CPU-Nutzungshistorien-Managementtabelle **1001** und die Speichernutzungshistorien-Managementtabelle **1101** sind in dem nichtflüchtigen Bereich **114** gespeichert. Die Tabellen **1001** und **1101** geben die größte und die kleinste Betriebsmittelmengen, die jedem Gast-OS der CPU zugewiesen ist, und den Speicher und die Gesamtmengen (Gesamtheiten) der zugewiesenen Betriebsmittel an. Wenn eine Betriebsmittelplanung ausgeführt wird, nehmen der CPU-Scheduler **121** und der Speicher-Scheduler **122** auf die größte und auf die kleinste Betriebsmittelmengen

ge, die in den Tabellen **1001** und **1101** registriert sind, Bezug und halten die Zuweisung an, wenn eine Planung ausgeführt wird, die die Mengen übersteigt oder kleiner als sie ist. Da die zugewiesene Betriebsmittelmeng (Gesamtheit) für jeweilige Gast-OS gemanagt wird, ist es darüber hinaus möglich, den Betriebsmittelnutzungsdatensätzen entsprechendes Geld in Rechnung zu stellen.

[0085] Fig. 12 stellt einen Betriebsablauf dar, wenn der CPU-Scheduler **121** regulär startet.

[0086] In Schritt **1201** führt der CPU-Scheduler **121** einen Prozess des Aktualisierens der CPU-Nutzungshistorien-Managementtabelle **1001** aus. Genauer nimmt z. B. der CPU-Scheduler **121** auf die CPU-Zuweisungszustandstabelle **132** Bezug, um die jedem Gast-OS zugewiesene Anzahl von CPUs zu berechnen. Der CPU-Scheduler **121** addiert die berechnete Anzahl zugewiesener CPUs zu dem Wert des Felds **1014** der zugewiesenen CPU-Gesamtheit der CPU-Nutzungshistorien-Managementtabelle **1001** für jedes Gast-OS. Mit diesem Prozess ist es möglich, die Gesamtheit der CPU-Nutzungsdatensätze zu managen und der Gesamtheit entsprechendes Geld in Rechnung zu stellen.

[0087] Um die Beschreibung zu erleichtern, wird im Folgenden ein Ziel-Gast-OS – ein Gast-OS –, für das ein Betriebsmittel abgefertigt oder vorbelegt wird, als ein "erstes Gast-OS" bezeichnet.

[0088] In Schritt **1202** bestimmt der CPU-Scheduler **121** eine Planungsziel-CPU (eine physikalische CPU, für die abzufertigen oder vorzubelegen ist) des ersten Gast-OS. In diesem Fall spezifiziert der CPU-Scheduler **121** aus der CPU-Nutzungshistorien-Managementtabelle **1001** einen Bereich von CPU-Mengen (eine Kombination der Werte des Felds **1012** der kleinsten CPU-Menge und des Felds **1013** der größten CPU-Menge) des ersten Gast-OS **103**. Wenn ein Wert, der durch Addieren oder Subtrahieren der Menge der Planungsziel-CPU zu oder von dem Wert des Felds **1014** der zugewiesenen CPU-Gesamtheit entsprechend dem ersten Gast-OS **103** erhalten wird, die Obergrenze des spezifizierten Bereichs von CPU-Mengen übersteigt oder kleiner als die Untergrenze des spezifizierten Bereichs der CPU-Menge ist, hält der CPU-Scheduler **121** die Zuweisung an.

[0089] In Schritt **1203** vergleicht der CPU-Scheduler **121** den in Schritt **1202** bestimmten Planungszustand (eine Kombination des ersten Gast-OS **103** und der physikalischen CPU nach der Planung) mit der CPU-Zuweisungszustandstabelle **132**, um zu bestimmen, ob es eine Änderung der Zuweisung (eine Änderung der Kombination des ersten Gast-OS und der physikalischen CPU) gibt.

[0090] Falls es keine Änderung gibt, beendet der CPU-Scheduler **121** den Prozess in Schritt **1204**.

[0091] Falls es eine Änderung gibt, aktualisiert der CPU-Scheduler **121** die CPU-Zuweisungszustandstabelle **132** in Schritt **1205** (wobei genauer z. B. der dem ersten Gast-OS **103** entsprechende Wert des Zuweisungszweifelds **402** auf "Nr." aktualisiert wird).

[0092] In Schritt **1206** aktualisiert der CPU-Scheduler **121** die Anforderungsprotokolltabelle **131**. Genauer stellt z. B. der CPU-Scheduler **121** "CPU" für das Betriebsmittelfeld **301** ein, stellt er die Nummer der Planungsziel-CPU **111** des ersten Gast-OS **103** für das Nummernfeld **302** ein, stellt er die Nummern der Gast-OS, die die physikalische CPU **111** belegen sollen, bevor und nachdem die Planung ausgeführt wird, für die Felder **303** und **304** des alten bzw. des neuen Zustands ein und stellt er "in Bearbeitung" für das Verarbeitungszustandsfeld **306** ein.

[0093] In Schritt **1207** bestimmt der CPU-Scheduler **121**, ob die Anforderung, die an den CPU-Abfertigungsagenten **126** eingereicht werden soll, eine Abfertigungsanforderung oder eine Vorbelegungsanforderung ist. Falls die Anforderung eine Abfertigungsanforderung ist, geht der Ablauf zu Schritt **1209** über, und falls die Anforderung eine Vorbelegungsanforderung ist, geht der Ablauf zu Schritt **1208** über.

[0094] In Schritt **1208** sendet der CPU-Scheduler **121** eine Unterbrechungsbenachrichtigung an die Planungsziel-CPU **111**. Das erste Gast-OS **103** und der Hypervisor-Agent **102**, die in der Planungsziel-CPU **111** arbeiten, hören zu arbeiten auf, und daraufhin wird die Steuerung an den CPU-Abfertigungsagenten **130** übertragen. Darüber hinaus wird außerdem eine Stapelumschaltung ausgeführt und werden Operationen ausgeführt, die den CPU-Abfertigungsagent-Stapelbereich **811** als die Stapel verwenden. Daraufhin geht der Ablauf zu Schritt **1212** über.

[0095] In Schritt **1209** reicht der CPU-Scheduler **121** einen privilegierten Befehl zum Ausführen eines Cache/TLB-Flash-Prozesses (eines Cache/Übersetzungsassoziativpuffer-Flash-Prozesses) (genauer eines Prozesses zum Flashen von Daten in dem Cache und in dem TLB zu einem vorgegebenen Ablageziel (z. B. zu dem nichtflüchtigen Bereich **114**)) ein. Dieser privilegierte Befehl wird durch die aktuelle physikalische CPU **111** (die physikalische CPU **111**, die den Hypervisor **101** ausführt) empfangen und die physikalische CPU **111** führt den Cache/TLB-Flash-Prozess aus. Der Cache und der TLB können hier Bereiche in dem physikalischen Speicher **113** sein.

[0096] In Schritt **1210** sichert der CPU-Scheduler **121** den Ausführungskontext in dem ersten Kontextablagebereich **601**. Der Ausführungskontext speichert z. B. den Wert des Registers **115** der aktuellen

physikalischen CPU (**111**) (der physikalischen CPU **111**, die den Hypervisor **101** ausführt).

[0097] In Schritt **1211** reicht der CPU-Scheduler **121** einen Trap-Befehl ein. Im Ergebnis beginnt der CPU-Abfertigungsagent **130** in der physikalischen CPU **111**, die in Ausführung ist, zu arbeiten. In diesem Fall werden die zu verwendenden Stapelbereiche geändert und werden die nachfolgenden Prozesse des CPU-Abfertigungsagenten **130** unter Verwendung der Stapel des CPU-Abfertigungsagent-Stapelbereichs **811** bearbeitet.

[0098] In Schritt **1212** schließt der CPU-Scheduler **121** den Prozess ab.

[0099] Fig. 13 stellt einen Betriebsablauf des CPU-Schedulers **121** dar, der in Ansprechen auf einen Trap-Rücksprung von dem CPU-Abfertigungsagenten **126** gestartet wird.

[0100] In Schritt **1301** reicht der CPU-Scheduler **1301** einen hlt-Befehl ein. Wenn der hlt-Befehl eingereicht wird, führt der CPU-Scheduler **1301** eine Bereitschaft aus, bis die später beschriebene Neustartsteuerung **123** eine Unterbrechungsbenachrichtigung einreicht. Es besteht eine Möglichkeit, dass der Hypervisor **101** wegen eines Fehlers des Hypervisors **101** neu gestartet wird, während der CPU-Abfertigungsagent **126** den Abfertigungs-/Vorbelegungsprozess für die physikalische CPU **111** als ein Agent ausführt. In diesem Fall wird der in Schritt **1210** gesicherte Kontext ungültig. Durch Ausführen der Bereitschaft, bevor der Kontext wiedergewonnen wird, ist es möglich, die Verwendung des ungültigen Kontexts zu beseitigen und die weiteren Operationen zu ermöglichen, nachdem der Hypervisor **101** neu gestartet ist.

[0101] In Schritt **1302** gewinnt der CPU-Scheduler **121** den Kontext aus dem Hypervisor-Kontextablagebereich **301** wieder. In diesem Fall wird der jeweilige Wert des Kontexts in dem Bereich **301** in das Register **115** der physikalischen CPU **111** (der physikalischen CPU **111**, die den Hypervisor **101** ausführt) geladen. Danach werden Operationen durch den Hypervisor **101** und durch das zweite Gast-OS **104** ausgeführt, während die physikalische CPU exklusiv belegt ist.

[0102] In Schritt **1303** schließt der CPU-Scheduler **121** den Prozess ab.

[0103] Allerdings beginnt der CPU-Abfertigungsagent **126** zu arbeiten, wenn durch reguläre Bezugnahme auf die Anforderungsprotokolltabelle **131** eine Abfertigungsanforderung detektiert wird (der Fall einer Abfertigung) oder eine Unterbrechungsbenachrichtigung von dem CPU-Scheduler **129** empfangen wird (der Fall einer Vorbelegung). Wenn eine CPU-Abfertigungsanforderung detektiert wird, führt

der CPU-Abfertigungsagent **126** den in Fig. 14 dargestellten Betriebsablauf aus. Wenn eine CPU-Vorbelegungsanforderung detektiert wird, führt der CPU-Abfertigungsagent **126** den Prozess des in Fig. 15 dargestellten Betriebsablaufs aus.

[0104] Fig. 14 stellt einen Betriebsablauf des Gast-CPU-Schedulers **129** und des CPU-Abfertigungsagenten **126**, der eine CPU-Abfertigungsanforderung detektiert hat, dar.

[0105] In Fig. 14 gewinnt der CPU-Abfertigungsagent **126** in Schritt **1401** den Kontext aus dem zweiten Kontextablagebereich **701** wieder. Genauer lädt z. B. der CPU-Abfertigungsagent **126** den jeweiligen Wert des Kontexts in dem zweiten Kontextablagebereich **701** in das Register **115** der physikalischen CPU **111**. Die physikalische CPU **111** ist die physikalische CPU **111**, die den Hypervisoragenten **102** ausführt (d. h. die physikalische CPU **111**, die das erste Gast-OS **103** ausführt).

[0106] In Schritt **1402** reicht der CPU-Abfertigungsagent **126** eine CPU-Aktualisierungsanforderung an den Gast-CPU-Scheduler **129** des ersten Gast-OS **103** ein, die eine Anforderung zum Aktualisieren von CPU-Konfigurationsinformationen ist.

[0107] In Schritt **1403** empfängt der Gast-OS-Scheduler **129** die CPU-Aktualisierungsanforderung. In diesem Fall wird die Nummer der physikalischen CPU **111** des Abfertigungsziels dem Gast-OS-Scheduler **129** mitgeteilt. Genauer kann z. B. die CPU-Aktualisierungsanforderung die Nummer der physikalischen CPU **111** des Abfertigungsziels enthalten und kann die Nummer der physikalischen CPU **111** des Abfertigungsziels dem Gast-OS-Scheduler **129** zusammen mit der Anforderung mitgeteilt werden.

[0108] In Schritt **1404** fügt der Gast-CPU-Scheduler **129** die mitgeteilte Nummer der physikalischen CPU zu einer Planungsziel-CPU-Liste des ersten Gast-OS **103** hinzu. Danach kann das erste Gast-OS **103** die physikalische CPU **111** verwenden.

[0109] In Schritt **1405** sendet der Gast-CPU-Scheduler **129** eine Benachrichtigung über den Abschluss der CPU-Aktualisierung an den CPU-Abfertigungsagenten **126**.

[0110] In Schritt **1406** empfängt der CPU-Abfertigungsagent **126** die Benachrichtigung über den Abschluss.

[0111] In Schritt **1407** aktualisiert der CPU-Abfertigungsagent **126** die Anforderungsprotokolltabelle **131** (wobei genauer z. B. der Wert des Verarbeitungszustandsfelds **306**, der der detektierten Abfertigungsanforderung entspricht, auf "abgeschlossen" aktualisiert wird).

[0112] In Schritt **1408** schließt der CPU-Abfertigungsagent **126** den Prozess ab.

[0113] **Fig. 15** stellt einen Betriebsablauf des Gast-CPU-Schedulers **129** und des CPU-Abfertigungsagenten **126**, der eine CPU-Vorbelegungsanforderung detektiert hat, dar.

[0114] In Schritt **1501** sendet der CPU-Abfertigungsagent **126** eine CPU-Aktualisierungsanforderung an den Gast-CPU-Scheduler **129** des ersten Gast-OS **103**.

[0115] In Schritt **1502** empfängt der Gast-CPU-Scheduler **129** die CPU-Aktualisierungsanforderung. In diesem Fall wird die Nummer der physikalischen CPU **111** des Verarbeitungsziels dem Gast-CPU-Scheduler **129** in Übereinstimmung mit demselben Verfahren wie in **Fig. 14** mitgeteilt.

[0116] In Schritt **1503** löscht der Gast-CPU-Scheduler **129** die mitgeteilte Nummer der physikalischen CPU aus der Planungsziel-CPU-Liste des ersten Gast-OS **103**. Danach kann das erste Gast-OS **103** die physikalische CPU **111** nicht verwenden.

[0117] In Schritt **1504** sendet der Gast-CPU-Scheduler **129** eine Benachrichtigung über den Abschluss der CPU-Aktualisierung an den CPU-Abfertigungsagenten **126**.

[0118] In Schritt **1505** empfängt der CPU-Abfertigungsagent **126** die Benachrichtigung über den Abschluss.

[0119] In Schritt **1506** aktualisiert der CPU-Abfertigungsagent **126** die Anforderungsprotokolltabelle **131** (wobei genauer z. B. der Wert des Verarbeitungszustandsfelds **206**, der der detektierten CPU-Vorbelegungsanforderung entspricht, auf "abgeschlossen" aktualisiert wird).

[0120] In Schritt **1507** sichert der CPU-Abfertigungsagent **126** den Kontext in dem zweiten Kontextablagebereich **701**. Genauer speichert z. B. der CPU-Abfertigungsagent **126** den in dem Register **115** der physikalischen CPU **111** gespeicherten Wert in dem zweiten Kontextablagebereich **701**. Die physikalische CPU **111** ist die physikalische CPU **111**, die den Hypervisoragenten **102** ausführt (d. h. die physikalische CPU **111**, die das erste Gast-OS **103** ausführt).

[0121] In Schritt **1508** reicht der CPU-Abfertigungsagent **126** einen Trap-Rücksprungbefehl (siehe **Fig. 8**) ein. Wenn der Trap-Rücksprungbefehl eingeht, werden Stapel umgeschaltet und arbeitet der arbeitende CPU-Scheduler **121** nachfolgend unter Verwendung des CPU-Scheduler-Stapelbereichs **815**.

[0122] In Schritt **1509** wird der Prozess abgeschlossen.

[0123] **Fig. 16** stellt einen Betriebsablauf des Speicher-Schedulers **122** dar. Der Speicher-Scheduler arbeitet regulär.

[0124] In Schritt **1601** führt der Speicher-Scheduler **122** einen Prozess zum Aktualisieren der Speichernutzungshistorien-Managementtabelle **1101** aus. Genauer nimmt z. B. der Speicher-Scheduler **122** auf die Speicher-Zuweisungszustandstabelle **133** Bezug, um die Menge des jedem Gast-OS zugewiesenen Speichers zu berechnen. Der Speicher-Scheduler **122** addiert die berechnete Menge des zugewiesenen Speichers zu dem Wert des Felds **1114** der Gesamtheit des zugewiesenen Speichers der Speichernutzungshistorien-Managementtabelle **1101** für jedes Gast-OS. Mit diesem Prozess ist es möglich, die Gesamtheit der Speichernutzungsdatensätze zu managen und entsprechend der Gesamtheit Geld in Rechnung zu stellen.

[0125] In Schritt **1602** bestimmt der Speicher-Scheduler **122** einen Planungszielspeicherbereich (einen Speicherbereich (eine Speicherseite), der (die) abzufertigen oder vorzubelegen ist) des ersten Gast-OS. In diesem Fall spezifiziert der Speicher-Scheduler **122** aus der Speichernutzungshistorien-Managementtabelle **1101** einen Bereich von Speichermengen (eine Kombination der Werte des Felds **1111** der kleinsten Speichermenge und des Felds **1112** der größten Speichermenge) des ersten Gast-OS **103**. Wenn ein durch Addieren oder Subtrahieren der Menge des Planungszielspeicherbereichs zu oder von dem Feld **1114** der Gesamtheit des zugewiesenen Speichers, der dem ersten Gast-OS **103** entspricht, erhaltener Wert die Obergrenze des spezifizierten Bereichs von Speichermengen übersteigt oder kleiner als die Untergrenze des spezifizierten Bereichs von Speichermengen ist, hält der Speicher-Scheduler **122** die Zuweisung an.

[0126] In Schritt **1603** prüft der Speicher-Scheduler **122**, ob es eine Änderung der Menge des dem ersten Gast-OS zugewiesenen Speichers gibt.

[0127] Falls es keine Änderung gibt, schließt der Speicher-Scheduler **122** den Prozess in Schritt **1604** ab.

[0128] Falls es eine Änderung gibt, aktualisiert der Speicher-Scheduler **122** in Schritt **1605** die Speicher-Zuweisungszustandstabelle **133**. Wenn die eingereichte Anforderung eine Speicherabfertigung (eine Erhöhung der Menge des zugewiesenen Speichers) für das erste Gast-OS **103** erfordert, bestimmt der Speicher-Scheduler **122** einen Zuweisungsziel-Speicherbereich (eine physikalische Seite). Die physikalische Seite ist eine physikalische Seite, deren

Zuweisungsziel nicht vorhanden ist und die leer ist, oder eine Seite, die unter den dem zweiten Gast-OS **104** zugewiesenen physikalischen Seiten vorbelegt werden kann. Die Werte des Felds **502** der Nummer der physikalischen Seite des Gasts und des Zuweisungszielfelds **503** der Speicher-Zuweisungsstandstabelle **133** werden für die physikalische Seite auf "Nr." aktualisiert. Andererseits wird der Aktualisierungsprozess dieses Schritts nicht ausgeführt, wenn die Anforderung eine Vorbelegungsanforderung oder eine Adressenänderungsanforderung für das erste Gast-OS ist.

[0129] In Schritt **1606** führt der Speicher-Scheduler **122** einen Prozess des Registrierens von Daten in der Anforderungsprotokolltabelle **131** aus. Genauer aktualisiert z. B. der Speicher-Scheduler **122** die Werte des Betriebsmitteltypfelds **301** und des Verarbeitungszustandsfelds **306**, die der Anforderung entsprechen, auf "Speicher" bzw. "in Verarbeitung". Wenn die Anforderung eine Speicherabfertigungsanforderung für das erste Gast-OS **103** ist, gibt das Nummernfeld **302** eine Nummer der physikalischen Seite des Ziels an und geben die Felder **303** und **304** des alten und des neuen Zustands die Nummern der Gast-OS an, denen die physikalische Seite zugewiesen ist, bevor bzw. nachdem der Abfertigungsprozess ausgeführt wird. Wenn die Anforderung eine Speichervorbelegungsanforderung für das erste Gast-OS **103** ist, geben die Felder **303** und **304** des alten und des neuen Zustands die Nummern der Gast-OS an, denen die physikalische Seite zugewiesen ist, bevor bzw. nachdem der Abfertigungsprozess ausgeführt wird.

[0130] In Schritt **1607** wird der Prozess abgeschlossen.

[0131] Fig. 17 stellt einen Betriebsablaufplan des Speicherabfertigungsagenten **125** und des Gast-Speicher-Schedulers **128** dar.

[0132] In Schritt **1701** extrahiert der Speicherabfertigungsagent **125** aus der Anforderungsprotokolltabelle **131** eine Anforderung, deren Wert des Quelltypfelds **301** "Speicher" ist und deren Wert des Verarbeitungszustandsfelds **306** "in Verarbeitung" ist.

[0133] In Schritt **1702** identifiziert der Speicherabfertigungsagent **125** die extrahierte Anforderung (Spalte). Wenn das Nummernfeld **302**, die Felder **303** und **304** des alten und des neuen Zustands eingestellt worden sind, wird die Anforderung als eine Speicherabfertigungsanforderung identifiziert. Wenn die Felder **303** und **304** des alten und des neuen Zustands eingestellt worden sind, das Nummernfeld **302** aber nicht eingestellt worden ist, wird die Anforderung als eine Speichervorbelegungsanforderung identifiziert. Wenn das Nummernfeld **302** und das Feld **304** des neuen Zustands eingestellt worden sind,

das Feld **303** des alten Zustands aber nicht eingestellt worden ist, wird die Anforderung als eine Adressenänderungsanforderung identifiziert. Wenn die Anforderung eine Abfertigungs-/Vorbelegungsanforderung ist, geht der Ablauf zu Schritt **1703** über, und wenn die Anforderung eine Adressenänderungsanforderung ist, geht der Ablauf zu Schritt **1708** über.

[0134] In Schritt **1703** reicht der Speicherabfertigungsagent **125** eine Speicheraktualisierungsanforderung, die eine Anforderung zum Aktualisieren von Speicherkonfigurationsinformationen ist, an den Gast-Speicher-Scheduler **128** des ersten Gast-OS ein.

[0135] In Schritt **1704** empfängt der Gast-Speicher-Scheduler **128** die Anforderung.

[0136] In Schritt **1705** fügt der Gast-Speicher-Scheduler **128** in Ansprechen auf die Anforderung zu einer Liste physikalischer Seiten, die eine Liste von Nummern physikalischer Seiten des Gasts ist, die das erste Gast-OS verwenden kann, eine Seitennummer hinzu und löscht er eine. In diesem Fall bestimmt der Gast-Speicher-Scheduler **128** die Nummer der physikalischen Seite eines Gasts, die hinzuzufügen und zu löschen ist. Falls die Anforderung eine Speicherabfertigungsanforderung ist, bestimmt der Gast-Speicher-Scheduler **128** die Nummer der physikalischen Seite eines Gasts, die zu der Liste hinzuzufügen ist. Falls die Anforderung eine Speichervorbelegungsanforderung ist, bestimmt der Gast-Speicher-Scheduler **128** die Nummer der physikalischen Seite eines Gasts, die aus der Liste zu löschen ist.

[0137] In Schritt **1706** sendet der Gast-Speicher-Scheduler **128** eine Benachrichtigung über den Abschluss der Speicheraktualisierung an den Speicherabfertigungsagenten **125**. In diesem Fall wird die in Schritt **1705** bestimmte Nummer der physikalischen Seite des Gasts den Speicherabfertigungsagenten **125** mitgeteilt. Die Nummer der physikalischen Seite des Gasts kann in der Anforderung enthalten sein und kann getrennt von der Benachrichtigung über den Abschluss (der Antwort) mitgeteilt werden.

[0138] In Schritt **1707** empfängt der Speicherabfertigungsagent **125** die Benachrichtigung über den Abschluss und die Nummer der physikalischen Seite des Gasts.

[0139] In Schritt **1708** aktualisiert der Speicherabfertigungsagent **125** die Adressenumsetzungstabelle **134**. Wenn die Anforderung eine Speicherabfertigungsanforderung ist, registriert der Speicherabfertigungsagent **125** eine Gast-OS-Nummer, die durch das der Anforderung entsprechende Feld **304** des neuen Zustands angegeben wird, die in Schritt **1707** empfangene Nummer der physikalischen Seite des Gasts und den Wert des Felds **603** der Nummer der

physikalischen Seite des Hosts, der durch das der Anforderung entsprechende Nummernfeld **302** angegeben wird, als die Werte des Felds **902** der Nummer des Gast-OS, des Felds **903** der Nummer der physikalischen Seite des Gasts und des Felds **904** der Nummer der physikalischen Seite des Hosts der Adressenumsetzungstabelle **134** und aktualisiert er den Wert des V-Felds **901** in dem Eintrag, in dem diese Werte registriert sind, auf "1" (gültig). Wenn die Anforderung eine Speichervorbelegungsanforderung ist, ermittelt der Speicherabfertigungsagent **125** aus der Adressenumsetzungstabelle **134** einen Eintrag, der die Gast-OS-Nummer aufweist, die durch das der Anforderung entsprechende Feld **304** des neuen Zustands angegeben wird, und die in Schritt **1707** empfangene Nummer der physikalischen Seite des Gasts in dem Feld **902** der Nummer des Gast-OS bzw. in dem Feld **903** der Nummer der physikalischen Seite des Gasts, und aktualisiert er den Wert des V-Felds **901** des ermittelten Eintrags auf "0" (ungültig). Ferner aktualisiert der Speicherabfertigungsagent **125** die Werte des Felds **502** der Nummer der physikalischen Seite des Gasts und des Zuweisungszielfelds **503** (siehe Fig. 5) entsprechend dem Feld **904** der Nummer der physikalischen Seite des Hosts in dem ermittelten Eintrag auf "Nr.". Wenn die Anforderung eine Adressenänderungsanforderung ist, registriert der Speicherabfertigungsagent **125** die Gast-OS-Nummer, die durch das der Anforderung entsprechende Feld **304** des neuen Zustands angegeben wird, die Nummer der physikalischen Seite des Gasts, die durch das der Anforderung entsprechende Feld **305** der Nummer der physikalischen Seite des Gasts angegeben wird, und den Wert des Felds **603** der Nummer der physikalischen Seite des Hosts, der durch das der Anforderung entsprechende Nummernfeld **302** angegeben wird, in dieser Reihenfolge als die Werte des Felds **902** der Nummer des Gast-OS, des Felds **903** der Nummer der physikalischen Seite des Gasts und des Felds **904** der Nummer der physikalischen Seite des Hosts der Adressenumsetzungstabelle **134** und aktualisiert er den Wert des V-Felds **901** in dem Eintrag, in dem diese Werte registriert sind, auf "1" (gültig).

[0140] In Schritt **1709** reicht der Speicherabfertigungsagent **125** einen privilegierten Befehl zum Ausführen eines Cache/TLB-Säuberungsprozesses – genauer einer Cache-/TLB-Säuberung (eines Prozesses des Löschsens von Daten in dem Cache und in dem TLB) ein. Dieser privilegierte Befehl wird von der physikalischen CPU **111** (der physikalischen CPU **111**, die den Hypervisor **10** ausführt) empfangen und die physikalische CPU **111** führt den Cache/TLB-Säuberungsprozess aus. Der Cache und der TLB können hier Bereiche in dem physikalischen Speicher **112** (Schutzbereich) sein.

[0141] In Schritt **1710** aktualisiert der Speicherabfertigungsagent **125** die Anforderungsprotokolltabelle

131. Genauer aktualisiert der Speicherabfertigungsagent **125** z. B. den der Anforderung entsprechenden Wert des Verarbeitungszustandsfelds **306** auf "abgeschlossen". Darüber hinaus aktualisiert der Speicherabfertigungsagent **125** den Wert des Felds **305** der Nummer der physikalischen Seite des Gasts auf den in Schritt **1706** empfangenen Wert. Ferner wird der in Schritt **1708** gelesene Wert des Felds **904** der Nummer der physikalischen Seite des Hosts als der Wert des entsprechenden Nummernfelds **302** registriert, wenn die Anforderung eine Speichervorbelegungsanforderung ist.

[0142] In Schritt **1711** wird der Prozess abgeschlossen.

[0143] Fig. 18 stellt einen Betriebsablauf der Einheit **127** zur Ausführung privilegierter Befehle und des zustandslosen CPU-Emulators **124** dar.

[0144] In Schritt **1801** führt die Einheit **127** zur Ausführung privilegierter Befehle einen privilegierten Befehl aus. Im Ergebnis tritt bei der physikalischen CPU **111** ein Trap auf und wird die Steuerung an den zustandslosen CPU-Emulator **124** übertragen. Zum Beispiel weist eine physikalische CPU eines bestimmten Lieferanten sowohl eine Speicheremulationsfunktion als auch eine CPU-Emulationsfunktion auf. Allerdings sind alle diese Funktionen aktiviert oder deaktiviert und ist es nicht möglich, nur eine von beiden Funktionen zu aktivieren. Obwohl in der vorliegenden Ausführungsform eine Speicheremulationsfunktion einer physikalischen CPU als Mittel zur Verwirklichung der Speichervirtualisierung aufgerufen (aktiviert) wird, wird die CPU-Emulationsfunktion ebenfalls aktiviert, da es nicht möglich ist, nur die Speicheremulationsfunktion zu aktivieren. Dies ist ein Beispiel eines Falls, in dem ein privilegierter Befehl (ein der Ausführung zugeordneter Trap) auftritt.

[0145] In Schritt **1802** liest der zustandslose CPU-Emulator **124** Befehle in der Ausführung.

[0146] In Schritt **1803** führt der zustandslose CPU-Emulator **124** einen Prozess des Aktualisierens des physikalischen Speichers **112** und des Registers **115** der physikalischen CPU **111** (der physikalischen CPU **111**, die das erste Gast-OS **103** ausführt) in Übereinstimmung mit dem gelesenen Befehl aus.

[0147] In Schritt **1804** reicht der zustandslose CPU-Emulator **124** einen Trap-Rücksprungbefehl ein und kehrt die Steuerung zu der Einheit **127** zur Ausführung privilegierter Befehle zurück.

[0148] In Schritt **1805** setzt die Einheit **127** zur Ausführung privilegierter Befehle den Prozess fort und in Schritt **1806** schließt sie den Prozess ab.

[0149] Fig. 19 stellt den Ablauf eines regulären Betriebs der Neustartsteuerung **123** dar.

[0150] In Schritt **1901** ermittelt die Neustartsteuerung **123** aus der Anforderungsprotokolltabelle **131** einen Eintrag, dessen Verarbeitungszustandsfeld **306** "abgeschlossen" ist.

[0151] In Schritt **1902** bestimmt die Neustartsteuerung **123**, ob der Eintrag ermittelt worden ist. Falls der Eintrag vorhanden ist, geht der Ablauf zu Schritt **1903** über. Falls der Eintrag nicht vorhanden ist, geht der Ablauf zu Schritt **1906** über.

[0152] In Schritt **1903** aktualisiert die Neustartsteuerung **123** die CPU-Zuweisungszustandstabelle **132** und die Speicher-Zuweisungszustandstabelle **133**. Genauer aktualisiert die Neustartsteuerung **123** z. B. den Wert des Zuweisungszielfelds **402** entsprechend dem Nummernfeld **302** des Eintrags auf den durch das Feld **304** des neuen Zustands des Eintrags angegebenen Wert, wenn der in Schritt **1901** ermittelte Eintrag ein Eintrag ist, der einer CPU-Abfertigungs-/Vorbelegungsanforderung entspricht. Darüber hinaus aktualisiert die Neustartsteuerung **123** z. B. den Wert des Zuweisungszielfelds **502** in dem Eintrag entsprechend dem Nummernfeld **302** des Eintrags auf den durch das Feld **304** des neuen Zustands des Eintrags angegebenen Wert, wenn in Schritt **1901** ermittelt wird, dass der Eintrag ein Eintrag ist, der einer Speicherabfertigungsanforderung entspricht. Ferner aktualisiert die Neustartsteuerung **123** den Wert des Felds **502** der Nummer der physikalischen Seite des Gasts in dem Eintrag, der dem Nummernfeld **302** des Eintrags entspricht, auf den durch das Nummernfeld **305** des physikalischen Gasts des Eintrags entsprechenden Wert. Darüber hinaus aktualisiert die Neustartsteuerung **123** die Werte des Felds **502** der Nummer der physikalischen Seite des Gasts und des Zuweisungszielfelds **503** in dem Eintrag entsprechend dem Feld **302** der Nummer des Eintrags auf "F", wenn der in Schritt **1901** ermittelte Eintrag ein Eintrag ist, der einer Speichervorbelegungsanforderung entspricht. Darüber hinaus kann die Neustartsteuerung **123** z. B. überhaupt nichts ausführen, wenn der in Schritt **1901** ermittelte Eintrag ein Eintrag ist, der einer Adressenaktualisierungsanforderung entspricht. Nach Ausführen des Aktualisierungsprozesses löscht die Neustartsteuerung **123** den Eintrag (den ermittelten Eintrag) aus der Anforderungsprotokolltabelle **131**.

[0153] In Schritt **1904** bestimmt die Neustartsteuerung **123**, ob in dem in Schritt **1901** ermittelten Eintrag eine CPU-Vorbelegungsabschlussanforderung enthalten ist. Falls die Anforderung enthalten ist, geht der Ablauf zu Schritt **1905** über, und falls die Anforderung nicht enthalten ist, geht der Ablauf zu Schritt **1906** über.

[0154] In Schritt **1905** sendet die Neustartsteuerung **123** einen Unterbrechungsbefehl an die physikalische Ziel-CPU **111**. Durch Senden dieses Unterbrechungsbefehls kann der Betrieb des CPU-Schedulers **121**, der in Schritt **1301** angehalten worden ist, sicher neu gestartet werden.

[0155] In Schritt **1906** wird der Prozess abgeschlossen.

[0156] Fig. 20 stellt einen Betriebsablauf der Neustartsteuerung **123** dar, nachdem der Hypervisor wegen eines Fehlers neu gestartet worden ist.

[0157] Wenn in Schritt **2001** in der Anforderungsprotokolltabelle **131** ein Eintrag vorhanden ist, dessen Wert des Verarbeitungszustandsfelds **306** "in Verarbeitung" ist, führt die Neustartsteuerung **123** eine Bereitschaft aus, bis der Verarbeitungszustand auf "abgeschlossen" geändert wird.

[0158] In Schritt **2002** bestimmt die Neustartsteuerung **123**, ob eine Bereitschaftszeitdauer abgelaufen ist. Wenn eine Zeitüberschreitung auftritt, geht der Ablauf zu Schritt **2003** über, und wenn keine Zeitüberschreitung auftritt, geht der Ablauf zu Schritt **2004** über.

[0159] In Schritt **2003** bricht die Neustartsteuerung **123** die Anforderung, deren Zeit überschritten wurde, ab. Genauer spezifiziert die Neustartsteuerung **123** z. B. den Wert des Nummernfelds **302** in einem Zieleintrag, der der Anforderung entspricht, deren Zeit überschritten wurde, und aktualisiert sie den Wert des Zuweisungszielfelds **402** entsprechend dem spezifizierten Nummernfeld **302** auf den Wert des Felds **303** des alten Zustands in dem Zieleintrag, wenn die Anforderung, deren Zeit überschritten wurde, eine CPU-Abfertigungs-/Vorbelegungsanforderung ist. Darüber hinaus spezifiziert die Neustartsteuerung **123** den Wert des Nummernfelds **302** in einem Zieleintrag, der der Anforderung entspricht, deren Zeit überschritten wurde, und aktualisiert sie die Werte des Felds **502** der Nummer der physikalischen Seite des Gasts und des Zuweisungszielfelds **503** entsprechend dem spezifizierten Nummernfeld **302** auf "F", wenn die Anforderung, deren Zeit überschritten wurde, eine Speicherabfertigungsanforderung ist. Wenn die Anforderung, deren Zeit überschritten wurde, eine Speichervorbelegungsanforderung oder eine Adressenänderungsanforderung ist, kann die Neustartsteuerung **123** nichts ausführen. Wenn der Abbruchprozess abgeschlossen ist, löscht die Neustartsteuerung **123** den Eintrag, der der Anforderung entspricht, deren Zeit überschritten wurde, aus der Anforderungsprotokolltabelle **131**.

[0160] In Schritt **2004** aktualisiert die Neustartsteuerung **123** den CPU-/Speicher-Zuweisungszustand. In diesem Aktualisierungsprozess werden ähnlich

dem Schritt **1903** die CPU-Zuweisungszustandstabelle **132** und die Speicher-Zuweisungszustandstabelle **133** in Übereinstimmung mit dem Registrierungsinhalt des Eintrags, dessen Wert des Verarbeitungszustandsfelds **306** der Anforderungsprotokolltabelle **131** "abgeschlossen" ist, aktualisiert.

[0161] In Schritt **2005** bestimmt die Neustartsteuerung **123** auf der Grundlage der Ergebnisse des Prozesses die CPU/den Speicher, die bzw. den der Hypervisor **101** und das zweite Gast-OS **104** verwenden können, und führt sie einen Prozess des Initialisierens (Bootens) des Hypervisors **101** und des zweiten Gast-OS **104** nur unter Verwendung der verfügbaren CPU/des verfügbaren Speichers (ohne Verwendung der Betriebsmittel, die der Hypervisoragent **102** und das erste Gast-OS **103** verwenden) aus. Auf diese Weise ist es möglich, den Hypervisor **101** und das zweite Gast-OS **104** neu zu starten, ohne den Hypervisoragenten **102** und das erste Gast-OS **103** zu beeinflussen.

[0162] In Schritt **2006** initialisiert die Neustartsteuerung **123** den ersten Kontextablagebereich **601**. Mit dieser Initialisierung ist garantiert, dass der Kontextwiedergewinnungsprozess des Schritts **1302** sicher ausgeführt wird.

[0163] Obwohl eine beispielhafte Ausführungsform zur Beschreibung der vorliegenden Erfindung beschrieben worden ist, soll der Schutzbereich der vorliegenden Erfindung nicht auf diese Ausführungsform beschränkt sein. Die vorliegende Erfindung kann in verschiedenen anderen Formen verkörpert werden.

Bezugszeichenliste

101	Hypervisor
102	Hypervisoragent
103	erstes Gast-OS
104	zweites Gast-OS
111	physikalische CPU
112	physikalischer Speicher (Schutzbereich)
113	physikalischer Speicher

Patentansprüche

1. Computer, der mehrere Betriebsmittel einschließlich eines oder mehrerer physikalischer Speicher und mehrerer physikalischer CPUs (Zentraleinheiten) umfasst, wobei die mehreren physikalischen CPUs dafür konfiguriert sind, einen Hypervisor, einen einfachen Hypervisor, ein erstes Gast-OS (Gast-Betriebssystem) und ein zweites Gast-OS auszuführen, unter den mehreren Betriebsmitteln ein von dem Hypervisor verwendetes Betriebsmittel von einem von dem einfachen Hypervisor verwendeten Betriebsmittel verschieden ist,

der eine oder die mehreren physikalischen Speicher einen ersten physikalischen Speicherbereich, der von dem Hypervisor nicht verwendet wird, und einen zweiten physikalischen Speicherbereich, der von dem Hypervisor verwendet wird, enthalten, das erste Gast-OS dafür konfiguriert ist, auf der Grundlage des ersten physikalischen Speicherbereichs in dem einfachen Hypervisor ausgeführt zu werden, das zweite Gast-OS dafür konfiguriert ist, auf der Grundlage des zweiten physikalischen Speicherbereichs in dem Hypervisor ausgeführt zu werden, der Hypervisor eine Emulationsfunktion, die eine Funktion zum Virtualisieren der mehreren Betriebsmittel als mehrere virtuelle Betriebsmittel ist, und eine Betriebsmittelplanungsfunktion, die eine Funktion zum Bestimmen von Betriebsmitteln, die sowohl dem ersten als auch dem zweiten Gast-OS dynamisch zuzuweisen sind oder deren Zuweisung zu ihnen dynamisch aufzuheben ist, enthält, der einfache Hypervisor eine Planungsagentenfunktion, die eine Funktion zum Zuweisen oder zum Aufheben der Zuweisung von Betriebsmitteln zu und von dem ersten Gast-OS in Übereinstimmung mit der Betriebsmittelplanungsfunktion des Hypervisors anstelle des Hypervisors ist, enthält, und der Hypervisor dafür konfiguriert ist, eine der Zuweisung oder der Aufhebung der Zuweisung von Betriebsmitteln zugeordnete Anforderung einzureichen und der einfache Hypervisor dafür konfiguriert ist, die Anforderung zu erfassen und einen Prozess in Übereinstimmung mit der Anforderung auszuführen.

2. Computer nach Anspruch 1, wobei die Anforderung in einen Ablagebereich geschrieben wird, und der einfache Hypervisor dafür konfiguriert ist, die Anforderung aus dem Ablagebereich zu erfassen.

3. Computer nach Anspruch 2, wobei eine zweite physikalische CPU, die von einer ersten physikalischen CPU, die den Hypervisor ausführt, verschieden ist, dafür konfiguriert ist, den einfachen Hypervisor auszuführen, und die Einreichung der Anforderung eine Einreichung eines Trap-Befehls oder eine Benachrichtigung über eine Unterbrechung der zweiten physikalischen CPU ist.

4. Computer nach Anspruch 2, wobei der Ablagebereich ein nichtflüchtiger Bereich ist, der Anforderungsprotokollinformationen speichert, die Anforderung als ein Protokoll zu den Anforderungsprotokollinformationen hinzugefügt wird, und der einfache Hypervisor dafür konfiguriert ist, das Protokoll in den Anforderungsprotokollinformationen auf der Grundlage der Ausführung eines der Anforderung entsprechenden Prozesses zu aktualisieren.

5. Computer nach Anspruch 1, wobei

der einfache Hypervisor dafür konfiguriert ist, den Betrieb in der physikalischen CPU eines Zuweisungsziels zu beginnen und das erste Gast-OS zu benachrichtigen, dass die physikalische CPU des Zuweisungsziels verfügbar geworden ist, wenn die Anforderung eine CPU-Zuweisungsanforderung ist, und der einfache Hypervisor dafür konfiguriert ist, das erste Gast-OS über die physikalischen CPU, die das Ziel der Aufhebung der Zuweisung ist, unter den dem ersten Gast-OS zugewiesenen physikalischen CPUs zu benachrichtigen und den Betrieb in der physikalischen CPU, die das Ziel der Aufhebung der Zuweisung ist, anzuhalten, wenn die Anforderung eine Anforderung zum Aufheben der Zuweisung der CPU ist.

6. Computer nach Anspruch 5, wobei die Einreichung der CPU-Zuweisungsanforderung die Einreichung eines Trap-Befehls ist, und die Einreichung der Anforderung zum Aufheben der Zuweisung der CPU eine Einreichung einer Benachrichtigung über eine Unterbrechung der physikalischen CPU, die nicht verfügbar geworden ist, ist.

7. Computer nach Anspruch 1, wobei der einfache Hypervisor dafür konfiguriert ist, Adressenumsetzungsinformationen, die eine Entsprechung zwischen einer Gastadresse, die eine logische Adresse ist, die von dem ersten Gast-OS bestimmt wird, und einer Host-Adresse, die eine physikalische Adresse eines Speicherbereichs des einen oder der mehreren physikalischen Speicher ist, zeigen, in der Weise zu aktualisieren, dass unter dem einen oder den mehreren physikalischen Speichern von dem ersten Gast-OS auf einen Zuweisungsziel-Speicherbereich zugegriffen werden kann und das erste Gast-OS benachrichtigt wird, dass der Zuweisungsziel-Speicherbereich verfügbar geworden ist, wenn die Anforderung eine Speicherzuweisungsanforderung ist, und der einfache Hypervisor dafür konfiguriert ist, das erste Gast-OS über einen nicht verwendbaren Speicherbereich unter den Speicherbereichen, die dem ersten Gast-OS zugeordnet sind, zu benachrichtigen und die Adressenumsetzungstabelle in der Weise aktualisieren, dass von dem ersten Gast-OS auf den nicht verwendbaren Speicherbereich nicht zugegriffen werden kann, wenn die Anforderung eine Anforderung zum Aufheben der Speicherzuweisung ist.

8. Computer nach Anspruch 4, wobei jedes Protokoll in den Anforderungsprotokollinformationen einen Fortschrittzustand eines Prozesses in Übereinstimmung mit einer Anforderung, die dem Protokoll entspricht, enthält, wobei der nichtflüchtige Bereich dafür konfiguriert ist, ferner Zuweisungszustandsinformationen zu speichern, die einen Zuweisungszustand der mehreren Betriebsmittel angeben, und der Hypervisor dafür konfiguriert ist, auf die Anforderungsprotokollinformationen und auf die Zuwei-

sungszustandsinformationen in dem nichtflüchtigen Bereich Bezug zu nehmen, eine Bereitschaft auszuführen, bis ein Prozess für eine Anforderung in Verarbeitung abgeschlossen ist, die Zuweisungszustandsinformationen auf der Grundlage der Anforderungsprotokollinformationen zu aktualisieren und den Hypervisor und das zweite Gast-OS in Verwendung eines Betriebsmittels, das dem ersten Gast-OS unter den mehreren Betriebsmitteln auf der Grundlage der aktualisierten Zuweisungszustandsinformationen nicht zugewiesen ist, zu initialisieren, wenn der Hypervisor neu gestartet wird.

9. Computer nach Anspruch 1, wobei das erste Gast-OS, der einfache Hypervisor und die Adressenumsetzungsinformationen, die eine Entsprechung zwischen einer Gastadresse, die eine logische Adresse ist, die von dem ersten Gast-OS bestimmt ist, und einer Host-Adresse, die eine physikalische Adresse eines Speicherbereichs des einen oder der mehreren physikalischen Speicher ist, zeigen, in dem ersten Speicherbereich angeordnet sind, und der erste Speicherbereich ein Bereich ist, in den von dem Hypervisor und von dem zweiten Gast-OS keine Daten geschrieben werden können.

10. Computer nach Anspruch 1, wobei der einfache Hypervisor dafür konfiguriert ist, einen Wert eines Registers auf der Grundlage des privilegierten Befehls anstelle des Hypervisors zu aktualisieren, wenn von dem ersten Gast-OS ein privilegierter Befehl eingebracht wird.

11. Computer nach Anspruch 1, wobei der Hypervisor dafür konfiguriert ist, einen Bereich verwendbarer Betriebsmittelmengen für jedes Gast-OS zu spezifizieren, und der Hypervisor dafür konfiguriert ist, das Einreichen der Anforderung anzuhalten, wenn eine Zuweisung oder die Aufhebung einer Zuweisung von Betriebsmitteln in Übereinstimmung mit einer Zielanforderung ausgeführt wird und die Menge der zugewiesenen Betriebsmittel nicht zu dem spezifizierten Bereich gehört.

12. Computer nach Anspruch 1, wobei der Hypervisor dafür konfiguriert ist, eine Zuweisungsbetriebsmittelmengen, die eine Betriebsmittelmengen ist, die jedem Gast-OS zugewiesen ist, zu managen, und der Hypervisor dafür konfiguriert ist, die Zuweisungsbetriebsmittelmengen in Übereinstimmung mit der Zuweisung oder mit der Aufhebung der Zuweisung von Betriebsmitteln zu und von jedem Gast-OS zu aktualisieren.

13. Computer nach Anspruch 9, wobei der Hypervisor dafür konfiguriert ist, eine Adressenänderungsanforderung zum Steuern eines Speicher-

bereichs, auf den von dem zweiten Gast-OS zugegriffen werden kann, einzureichen, und der einfache Hypervisor dafür konfiguriert ist, die Adressenumsetzungsinformationen in Übereinstimmung mit der Adressenänderungsanforderung zu aktualisieren.

14. Computer nach Anspruch 4, wobei jedes Protokoll in den Anforderungsprotokollinformationen einen Fortschrittszustand eines Prozesses, der einer Anforderung entspricht, die dem Protokoll entspricht, enthält, der einfache Hypervisor dafür konfiguriert ist, das erste Gast-OS über eine physikalische CPU, die das Ziel der Aufhebung der Zuweisung ist, unter den physikalischen CPUs, die dem ersten Gast-OS zugewiesen sind, zu benachrichtigen und das Arbeiten in der physikalischen CPU, die das Ziel der Aufhebung der Zuweisung ist, anzuhalten, wenn die Anforderung eine Anforderung zum Aufheben der Zuweisung einer CPU ist, und der Hypervisor dafür konfiguriert ist, auf eine Benachrichtigung über eine Unterbrechung zu warten, nachdem der einfache Hypervisor den Betrieb in Übereinstimmung mit der Anforderung zum Aufheben der Zuweisung einer CPU angehalten hat, einen Fortschrittszustand eines Prozesses, der der Anforderung zum Aufheben der Zuweisung der CPU entspricht, aus der Referenzliste zu spezifizieren und die Benachrichtigung über die Unterbrechung der physikalischen CPU, die das Ziel der Aufhebung der Zuweisung ist, zu senden, wenn der spezifizierte Fortschrittszustand den Abschluss angibt.

15. Betriebsmittelplanungsverfahren zum Ausführen einer Betriebsmittelplanung mit Hilfe eines Hypervisors in einem Computer, der mehrere Betriebsmittel aufweist, die einen oder mehrere physikalische Speicher und mehrere physikalische CPUs (Zentraleinheiten) enthalten, wobei das Verfahren umfasst: Bereitstellen eines ersten physikalischen Speicherbereichs, der von dem Hypervisor nicht verwendet wird, und eines zweiten physikalischen Speicherbereichs, der von dem Hypervisor verwendet wird, in dem einen oder in den mehreren physikalischen Speichern und Ausführen eines einfachen Hypervisors in Verwendung eines Betriebsmittels, das von dem von dem Hypervisor verwendeten Betriebsmittel unter den mehreren Betriebsmitteln verschieden ist; Ausführen eines ersten Gast-OS (Gast-Betriebssystems) auf der Grundlage des ersten physikalischen Speicherbereichs in dem einfachen Hypervisor und Ausführen eines zweiten Gast-OS auf der Grundlage des zweiten physikalischen Speicherbereichs in dem Hypervisor, wobei der Hypervisor eine Emulationsfunktion, die eine Funktion zum Virtualisieren der mehreren Betriebsmittel als mehrere virtuelle Betriebsmittel ist, und eine Betriebsmittelplanungsfunktion, die eine Funktion zum Bestimmen von Betriebsmitteln, die dem ersten und dem zweiten Gast-OS dy-

namisch zuzuweisen sind oder deren Zuweisung dynamisch aufzuheben ist, aufweist und wobei der einfache Hypervisor eine Planungsagentenfunktion aufweist, die eine Funktion zum Zuweisen oder zum Aufheben der Zuweisung von Betriebsmitteln zu oder von dem ersten Gast-OS in Übereinstimmung mit der Betriebsmittelplanungsfunktion des Hypervisors anstelle des Hypervisors ist;

Veranlassen, dass der Hypervisor eine Anforderung zur Zuweisung oder Aufhebung der Zuweisung von Betriebsmitteln einreicht; und

Veranlassen, dass der einfache Hypervisor die Anforderung erfasst und einen Prozess in Übereinstimmung mit der Anforderung ausführt.

Es folgen 12 Seiten Zeichnungen

Anhängende Zeichnungen

FIG. 1

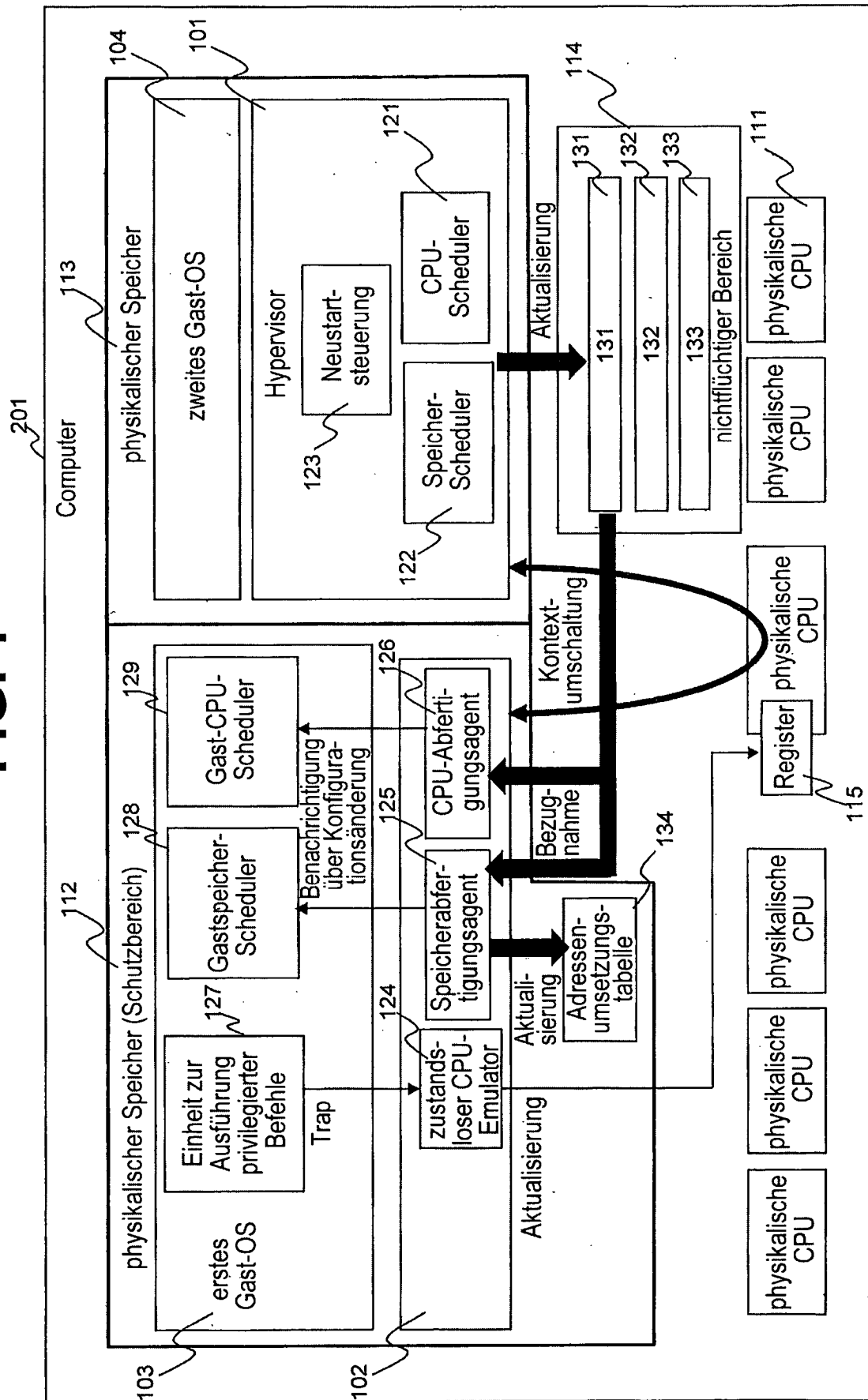


FIG. 2

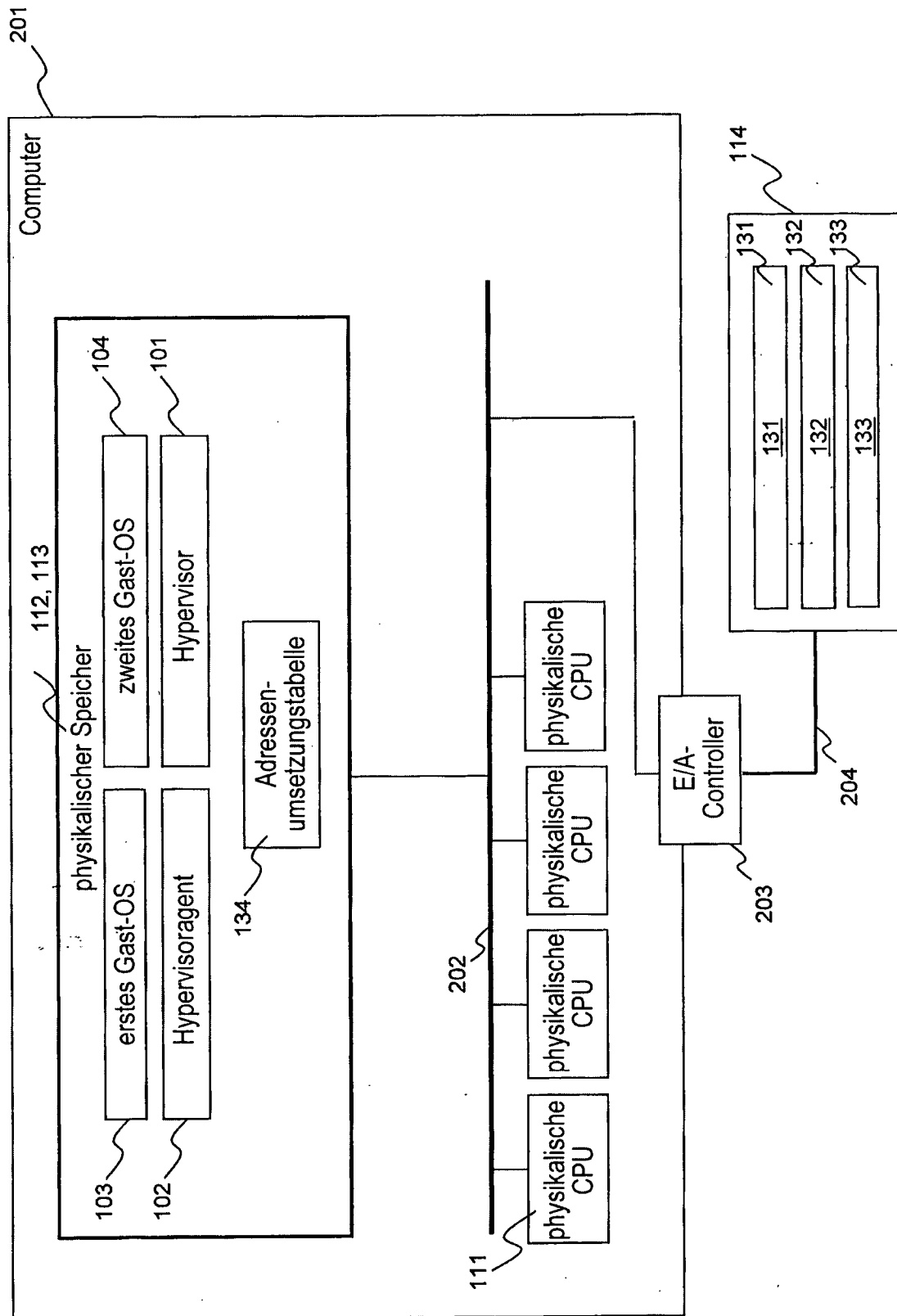


FIG. 3Anforderungsprotokolltabelle
131

Betriebsmitteltyp	CPU	Speicher	Speicher	Speicher	...	301
Nummer	2	4	-	5		302
alter Zustand	1	F	1	-		303
neuer Zustand	2	1	2	2		304
physikalische Nummer des Gasts	-	-	-	9		305
Verarbeitungszustand	abgeschlossen	in Verarbeitung	in Verarbeitung	in Verarbeitung	in Verarbeitung	306

FIG. 4CPU-Zuweisungszustandstabelle
132

CPU-Nummer	0	1	2	3	4	...	...	...	401
Zuweisungsziel	1	1	Nr.	F	2				402

FIG. 5Speicher-Zuweisungszustandstabelle
133

Nummer der physikalischen Seite des Hosts	0	1	2	3	4	...	...	...	501
Nummer der physikalischen Seite des Gasts	13	8	F	14	Nr.				502
Zuweisungsziel	1	2	F	1	Nr.				503

FIG. 6

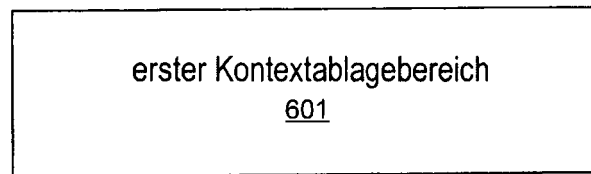


FIG. 7

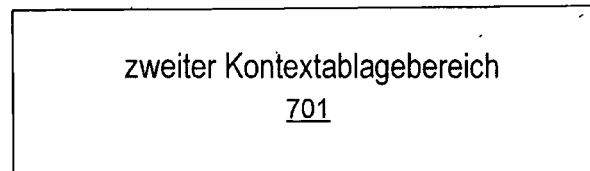


FIG. 8

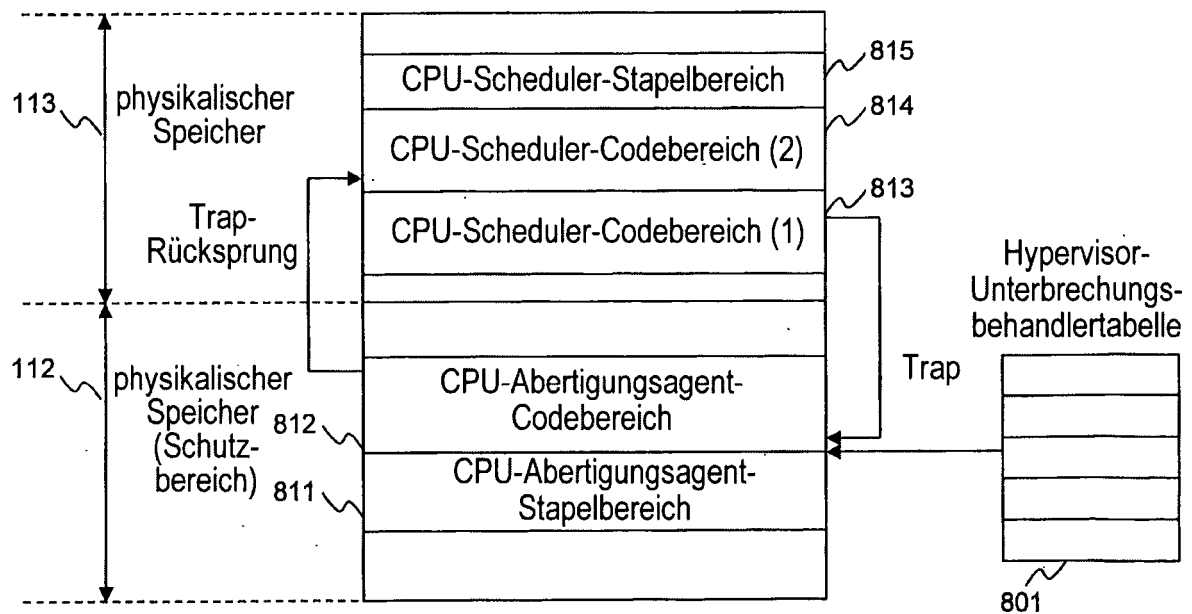


FIG. 9

Adressenumsetzungstabelle

134

V	Nummer des Gast-OS	Nummer der physikalischen Seite des Gasts	Nummer der physikalischen Seite des Hosts

FIG. 10

CPU-Nutzungshistorien-Managementtabelle

1001

Nummer des Gast-OS	kleinste CPU-Menge	größte CPU-Menge	zugewiesene CPU-Gesamtheit

FIG. 11

Speichernutzungshistorien-Managementtabelle

1101

Nummer des Gast-OS	kleinster Speicher	größte Speichermenge	zugewiesene Speichergesamtheit

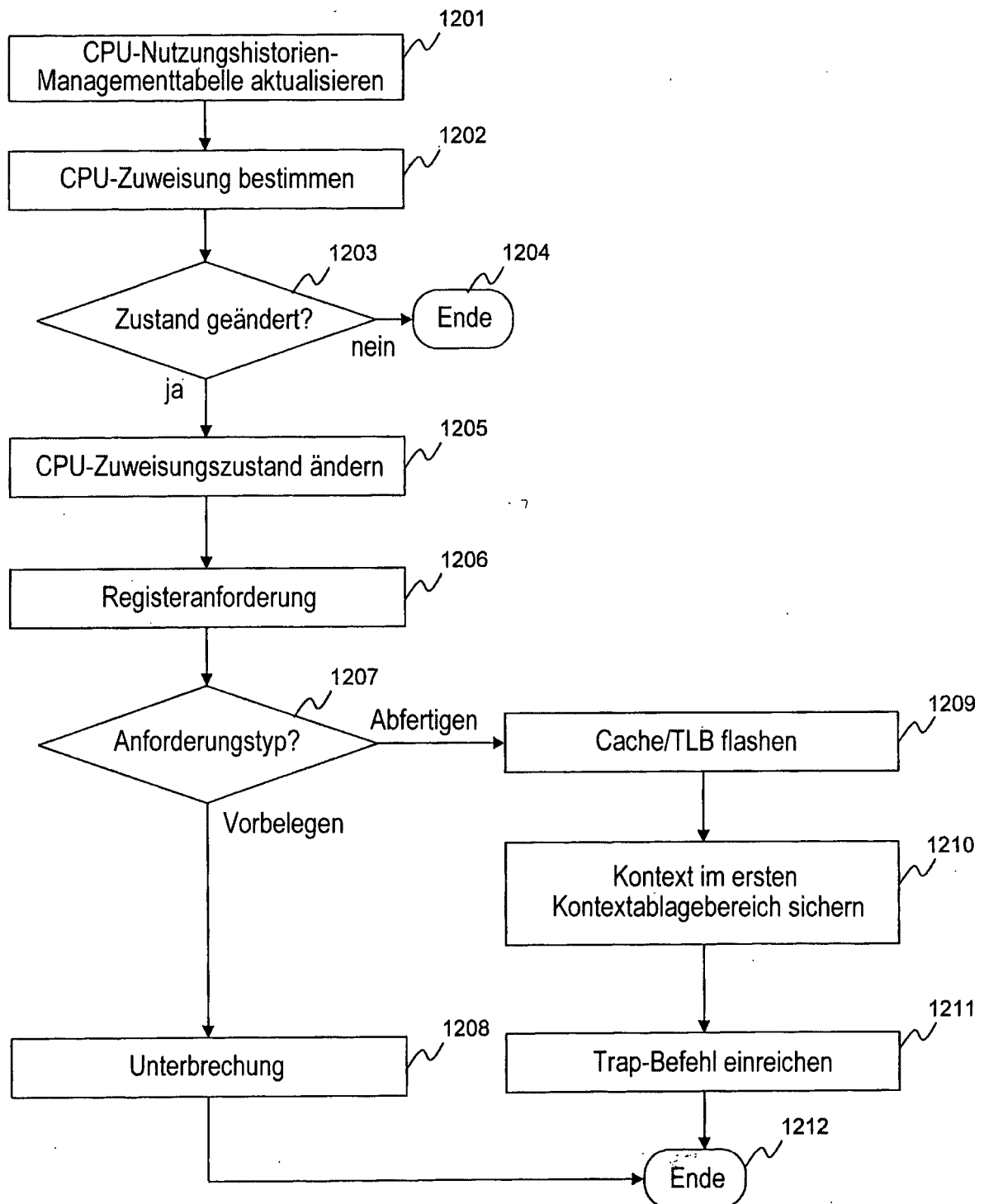
FIG. 12

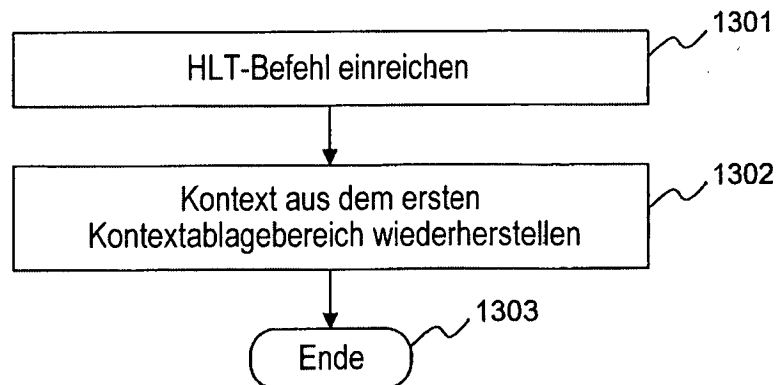
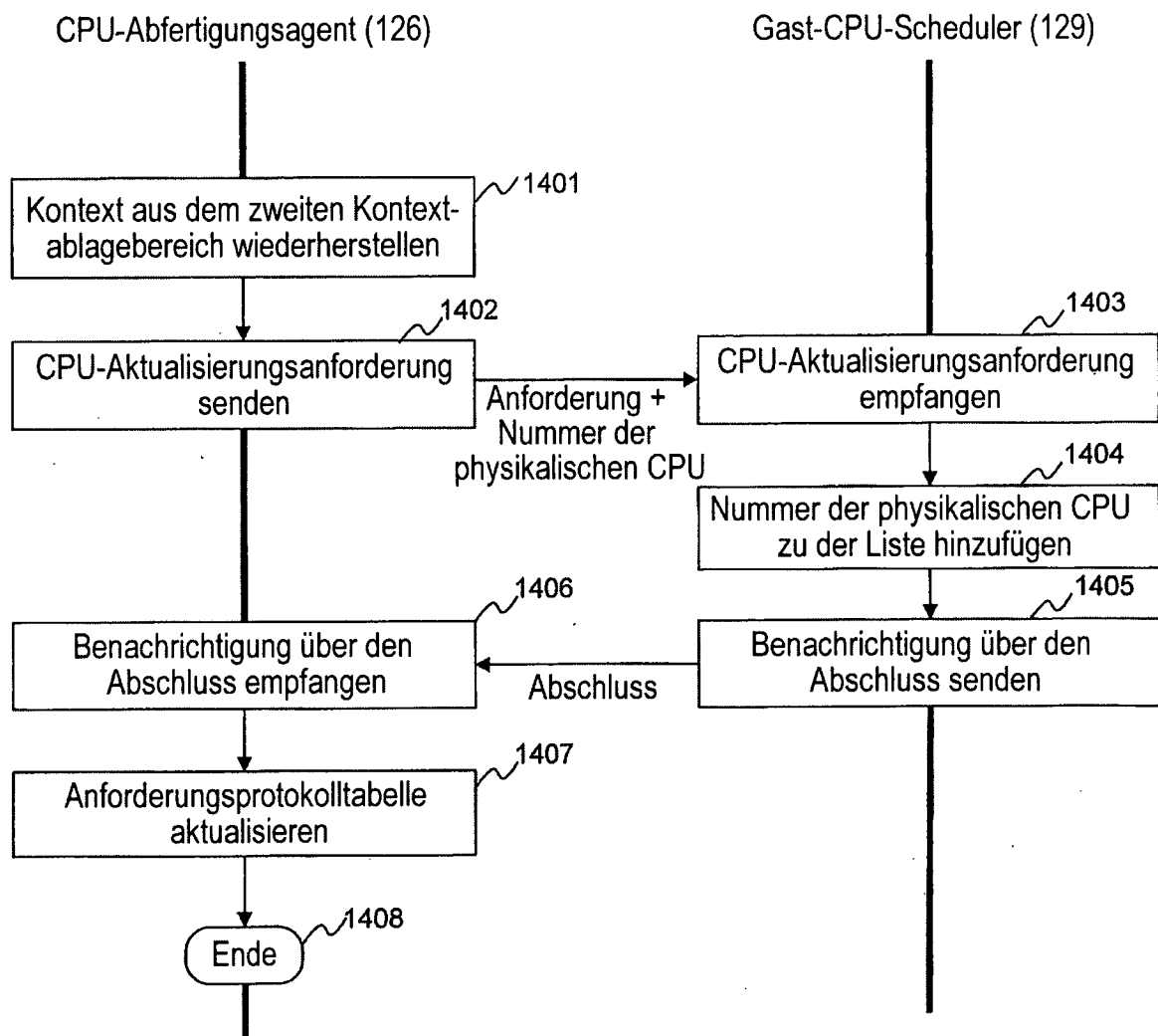
FIG. 13**FIG. 14**

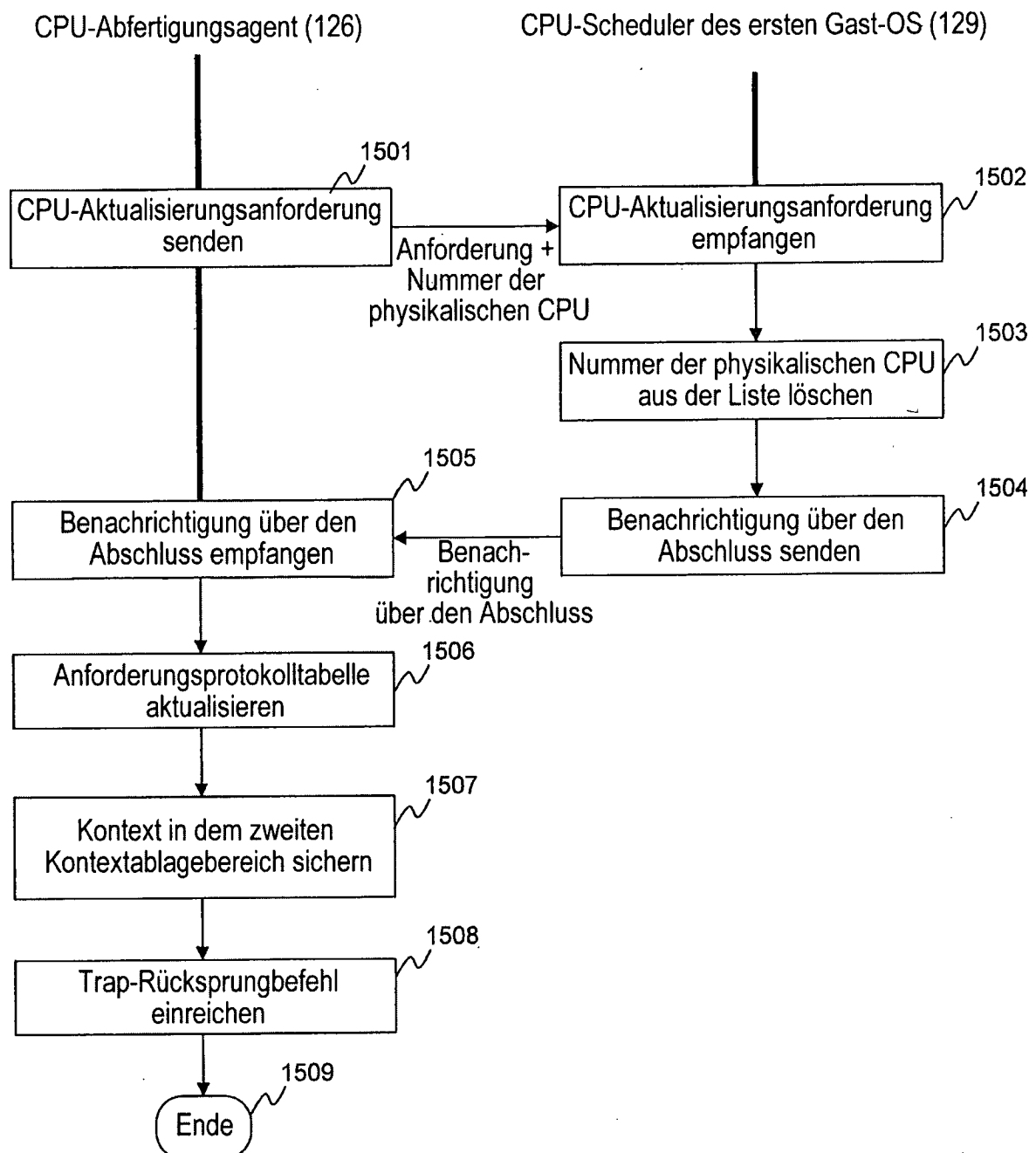
FIG. 15

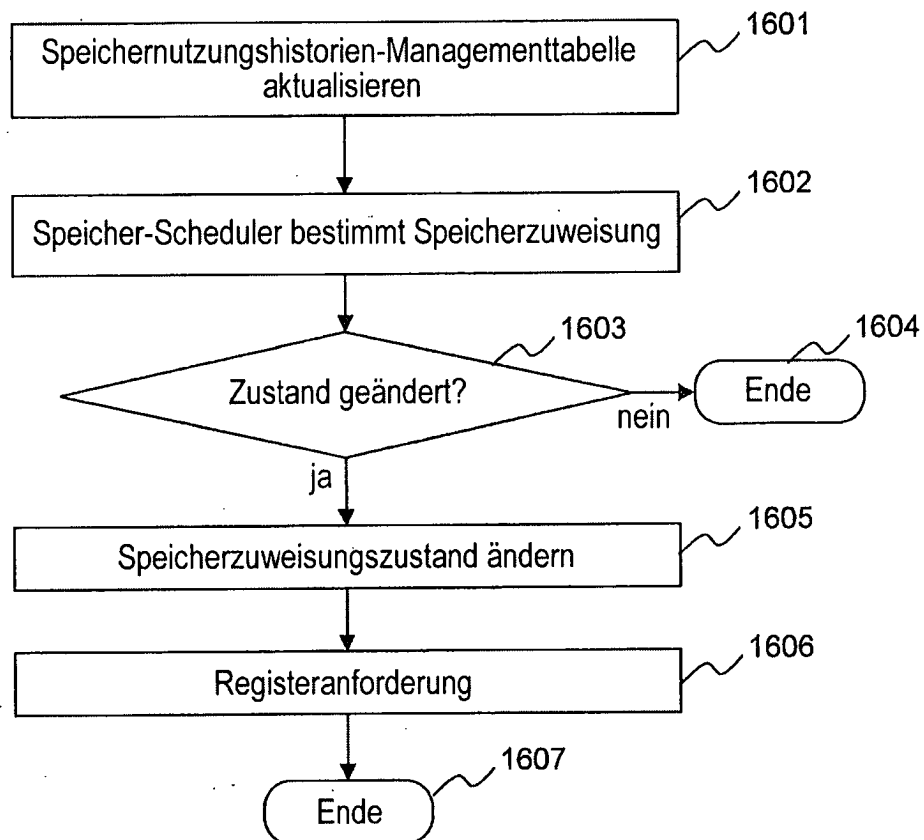
FIG. 16

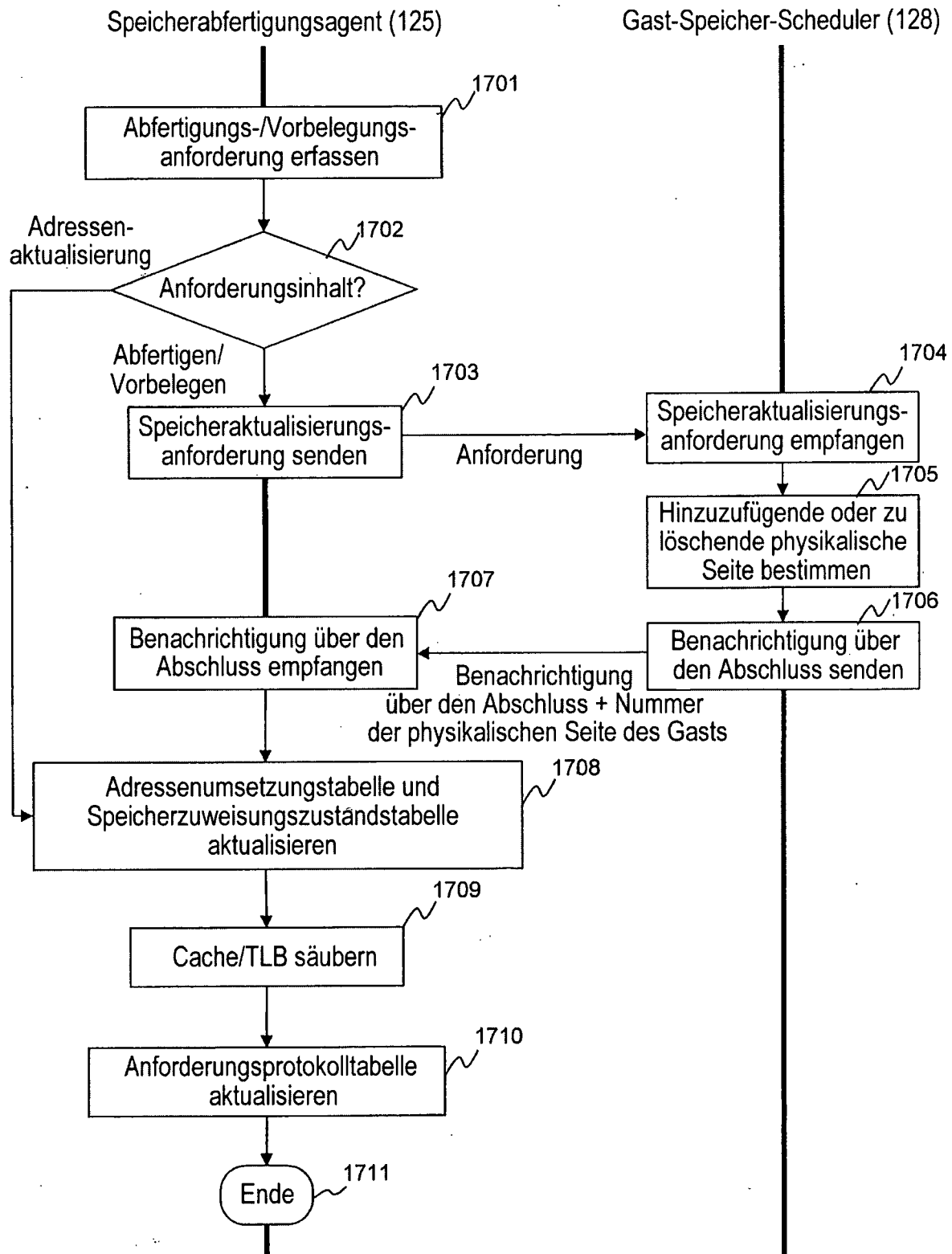
FIG. 17

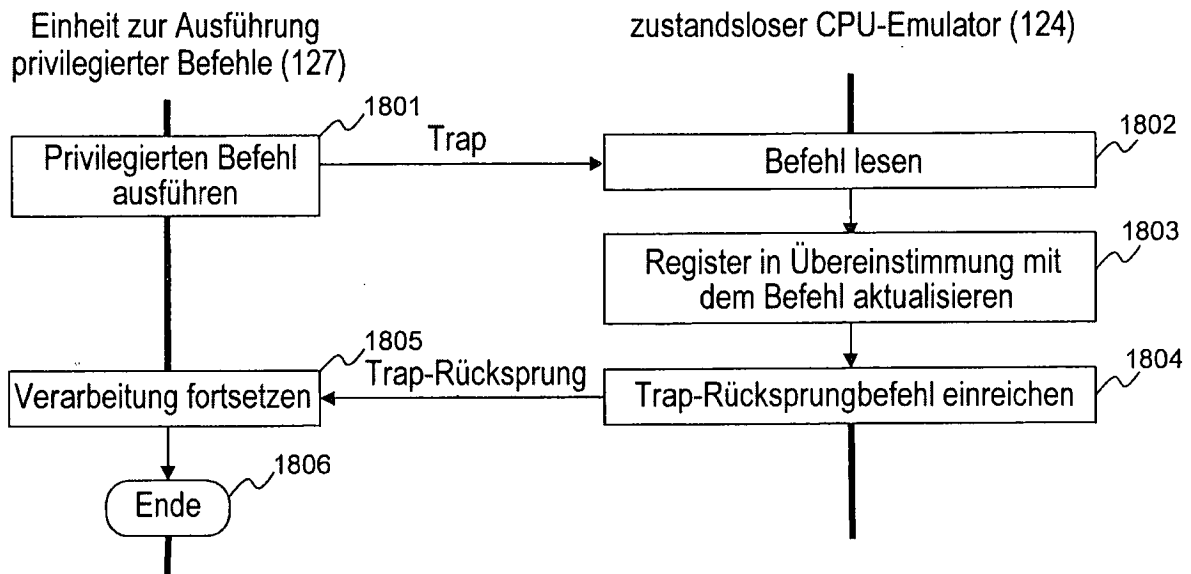
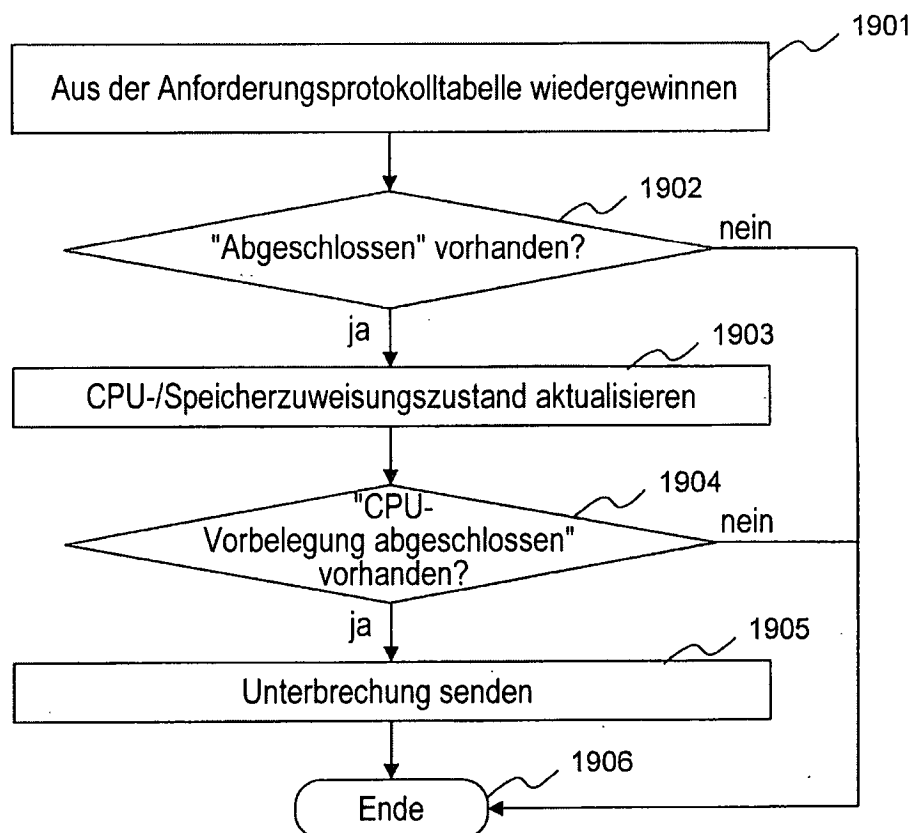
FIG. 18**FIG. 19**

FIG. 20