



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 603 18 687 T2** 2009.01.15

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 481 324 B1**

(51) Int Cl.⁸: **G06F 11/00** (2006.01)

(21) Deutsches Aktenzeichen: **603 18 687.4**

(86) PCT-Aktenzeichen: **PCT/US03/06620**

(96) Europäisches Aktenzeichen: **03 744 175.5**

(87) PCT-Veröffentlichungs-Nr.: **WO 2003/077128**

(86) PCT-Anmeldetag: **06.03.2003**

(87) Veröffentlichungstag
der PCT-Anmeldung: **18.09.2003**

(97) Erstveröffentlichung durch das EPA: **01.12.2004**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **16.01.2008**

(47) Veröffentlichungstag im Patentblatt: **15.01.2009**

(30) Unionspriorität:
90728 06.03.2002 US

(84) Benannte Vertragsstaaten:
DE, GB

(73) Patentinhaber:
**Marathon Technologies Corp., Boxborough,
Mass., US**

(72) Erfinder:
**TREMBLAY, Glenn A., Upton, MA 01568, US;
LEVEILLE, Paul A., Grafton, MA 01519, US;
KAMAN, Charles H., Lincoln, MA 01773, US;
GRANNUM, Gairy, Boxborough, MA 01719, US**

(74) Vertreter:
**Kuhnen & Wacker Patent- und
Rechtsanwaltsbüro, 85354 Freising**

(54) Bezeichnung: **HERSTELLEN EINER GESPIEGELTEN KOPIE UNTER VERWENDUNG INKREMENTELLER DI-
VERGENZ**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

TECHNISCHES GEBIET

[0001] Diese Erfindung betrifft Verfahren zum Erzeugen einer gespiegelten Kopie einer Festplatte oder einer anderen Speicherungseinrichtung.

HINTERGRUND

[0002] Bei vielen Computersystemen wird ein Maß an Fehlertoleranz vorgesehen, indem identische Daten auf jeder von mehreren Speicherungseinrichtungen gespeichert werden. Speicherungseinrichtungen mit identischen Daten werden als gespiegelte Einrichtungen bezeichnet und werden als zu einem Spiegelsatz gehörend betrachtet. Wenn eine gespiegelte Einrichtung in einem Spiegelsatz ausfällt oder anderweitig unzugänglich wird, sorgt die andere gespiegelte Einrichtung bzw. sorgen die anderen gespiegelten Einrichtungen weiterhin für Zugriff auf die Daten.

[0003] Um identische Daten auf jeder Einrichtung in einem Spiegelsatz zu unterhalten, muss jede Einrichtung jede Anforderung zum Speichern von Daten auf dem Spiegelsatz (d. h. jede Schreibanforderung) empfangen und verarbeiten. Eine Einrichtung in einem Spiegelsatz weicht von anderen Einrichtungen im Spiegelsatz ab, wenn die Einrichtung nicht in der Lage ist, solche Schreibanforderungen zu verarbeiten. Wenn Mitglieder eines Spiegelsatzes voneinander abweichen, kann eine Spiegelsatzkopie ausgeführt werden, um Daten von einer gespiegelten Einrichtung auf eine andere gespiegelte Einrichtung zu kopieren. Bei einer Vorgehensweise zum Unterhalten einer Spiegelsatzkopie wird das Computersystem ausgeschaltet und alle Daten werden von einer gespiegelten Einrichtung auf die andere gespiegelte Einrichtung kopiert.

[0004] US-Patent US6260125 offenbart ein asynchrones Plattenspiegelungssystem für den Einsatz in einem vernetzten Computersystem. Das Plattenspiegelungssystem umfasst ein erstes Speichervolumen, das angeschlossen ist, um Schreibanforderungen vom Computersystem zu empfangen; eine Schreibwarteschlange, die angeschlossen ist, um ebenfalls die an das erste Speichervolumen gerichteten Schreibanforderungen zu empfangen; und ein zweites Speichervolumen, das an die Schreibwarteschlange angeschlossen ist, um die Schreibanforderungen zu empfangen. Die Schreibwarteschlange ist wirksam, die Zeit des Empfangs der Schreibanforderungen durch das zweite Speichervolumen zu verzögern. Schreibanforderungen gelangen in einer First-in-first-out-Reihenfolge (FIFO) durch die Schreibwarteschlange, die mehrere seriell miteinander verbundene Schreibpuffer umfasst. Eine Protokolldatei, die dazu angeschlossen

ist, die verzögerten Schreibanforderungen von der Schreibwarteschlange zu empfangen, ist ebenfalls im Plattenspiegelungssystem enthalten, um für die protokollbasierte Spiegelrekonstruktion und die Ausführung einer Fixpunktoutine mit den gespiegelten Volumen zu sorgen.

[0005] Das dem Anmelder übertragene US-Patent 5,787,485 beschreibt und beansprucht ein Verfahren des Ausführens einer Spiegelsatzkopie von einer ersten Speicherungseinrichtung zu einer zweiten Speicherungseinrichtung, wobei Schreibanforderungen bei der ersten Speicherungseinrichtung empfangen und verarbeitet werden. Jede der Schreibanforderungen wird durch eine Referenzmarke identifiziert. Als Reaktion auf eine Spiegel-Leseanforderung werden Spiegeldaten von der ersten Speicherungseinrichtung gelesen und zur zweiten Speichervorrichtung geschickt, zusammen mit Informationen, die jede Schreibanforderung bezeichnen, die von der zweiten Speicherungseinrichtung verarbeitet werden kann. Die zur zweiten Speicherungseinrichtung geschickten Spiegeldaten werden in die zweite Speicherungseinrichtung geschrieben, die dann Schreibanforderungen gemäß der Bezeichnungsinformationen verarbeitet.

[0006] US-Patent 6,338,126 offenbart ein Absturzrettungssystem, das ein Primärcomputersystem und ein Sicherungscomputersystem nutzt. Jede Schreibanforderung wird an jedes Computersystem geschickt, wobei die Schreibanforderung an einen Verzögerungspuffer und eine Speicherwarteschlange des Primärcomputersystems und an eine Speicherwarteschlange des Sicherungscomputersystems geschickt wird. Der Sicherungscomputer überträgt eine Quittierung an den Primärcomputer, um dem Empfang der Schreibanforderung zu bestätigen, wonach die Schreibanforderung im Verzögerungspuffer des Primärcomputers ausgeführt wird. Der Sicherungscomputer führt die Schreibanforderung zu einer beliebigen Zeit aus. Falls einer der Computer abschaltet, werden die Schreibanforderungen in der Speicherwarteschlange des anderen Computers gesammelt.

ZUSAMMENFASSUNG

[0007] Die Erfindung ist im Einzelnen in den angehängten unabhängigen Patentansprüchen 1 und 32 definiert.

[0008] In einem allgemeinen Aspekt wird eine gespiegelte Kopie einer ersten Speicherungseinrichtung bei einer zweiten Speicherungseinrichtung in einem Computersystem unterhalten. Die erste Speicherungseinrichtung enthält einen assoziierten Controller und die zweite Speicherungseinrichtung enthält einen assoziierten Controller, flüchtige Speicherung und nichtflüchtige Speicherung. Bei den Speiche-

zungseinrichtungen empfangene Schreibanforderungen werden verarbeitet. Eine Commit-Synchronisierungs-Nachricht wird an die zweite Speicherungseinrichtung gesendet, zusammen mit Informationen, die eine Schreibanforderung bezeichnen, und der Controller der zweiten Speicherungseinrichtung bestätigt nach dem Empfangen der Commit-Synchronisierungs-Nachricht, dass die mit der bezeichneten Schreibanforderung assoziierten Daten in die nichtflüchtige Speicherung der zweiten Speicherungseinrichtung geschrieben wurden.

[0009] Umsetzungen können eines oder mehrere der folgenden Merkmale enthalten. Beispielsweise kann der Controller der zweiten Speicherungseinrichtung bestätigen, dass Daten, die mit allen Schreibanforderungen assoziiert sind, die der bezeichneten Schreibanforderung vorausgegangen sind, in die nichtflüchtige Speicherung der zweiten Speicherungseinrichtung geschrieben wurden. Alternativ kann der Controller der zweiten Speicherungseinrichtung die bezeichnete Schreibanforderung verarbeiten und kann bestätigen, dass Daten, die mit der bezeichneten Schreibanforderung und mit vorausgehenden Schreibanforderungen assoziiert sind, in die nichtflüchtige Speicherung der zweiten Speicherungseinrichtung geschrieben wurden. Der Controller der zweiten Speicherungseinrichtung kann eine erfolgreiche Cache-Leerung der flüchtigen Speicherung der zweiten Speicherungseinrichtung bestätigen.

[0010] Mit der Commit-Synchronisierungs-Nachricht gesendete Informationen enthalten eine Referenzmarke, die eine Schreibanforderung identifiziert, die von der ersten Speicherungseinrichtung verarbeitet wurde oder zu verarbeiten ist. Die Referenzmarke wird sequenziell relativ zu Referenzmarken zugeordnet, die anderen Schreibanforderungen zugeordnet sind. Alle bei der zweiten Speicherungseinrichtung empfangenen Schreibanforderungen können sequenziell verarbeitet werden, bevor die von der Referenzmarke in der Commit-Synchronisierungs-Nachricht identifizierte Schreibanforderung verarbeitet wird.

[0011] Die identifizierten Speicherungsgebiete, die von den Schreibanforderungen betroffen sind, können beispielsweise in einer ersten Bitmap akkumuliert werden. Nach dem Senden der Commit-Synchronisierungs-Nachricht können neu identifizierte Speicherungsgebieten in einer zweiten Bitmap akkumuliert werden. Nachdem der Controller der zweiten Speicherungseinrichtung bestätigt, dass Daten in den verarbeiteten Schreibanforderungen in nichtflüchtige Speicherung der zweiten Speicherungseinrichtung geschrieben wurden, kann eine Statusnachricht an die erste Speicherungseinrichtung gesendet werden, um anzuzeigen, dass die Schreibdaten erfolgreich in die nichtflüchtige Speicherung geschrie-

ben wurden. Nach dem Empfang der Statusnachricht, die anzeigt, dass die Schreibdaten erfolgreich geschrieben wurden, kann die erste Bitmap gelöscht werden und die zweite Bitmap kann als die erste Bitmap bezeichnet werden.

[0012] Nach einer Periode, in der die zweite Speicherungseinrichtung unverfügbar war, kann der Inhalt der ersten Bitmap in eine Rettungsbitmap kopiert werden, die dann verwendet wird, um die Speicherungsgebiete der ersten Speicherungseinrichtung zu identifizieren, die von der ersten Speicherungseinrichtung zur zweiten Speicherungseinrichtung kopiert werden sollen. Die identifizierten Speicherungsgebieten der ersten Speicherungseinrichtung können zur zweiten Speicherungseinrichtung kopiert werden und neu empfangene Schreibanforderungen können bei der zweiten Speicherungseinrichtung in einer dritten Bitmap akkumuliert werden.

[0013] Die zweite Speicherungseinrichtung kann eines oder mehrere der Merkmale und Funktionen ausführen, wie sie vorangehend anhand der ersten Speicherungseinrichtung beschrieben wurden und die erste Speicherungseinrichtung kann eines oder mehrere der Merkmale ausführen, die vorangehend anhand der zweiten Speichervorrichtung beschrieben wurden.

[0014] In einem weiteren allgemeinen Aspekt umfasst das Unterhalten einer gespiegelten Kopie einer ersten Speicherungseinrichtung bei einer zweiten Speicherungseinrichtung in einem Computersystem das Empfangen von Schreibanforderungen bei einer ersten Speicherungseinrichtung, die einen assoziierten Controller, flüchtige Speicherung und nichtflüchtige Speicherung enthält; das Verarbeiten der bei der ersten Speicherungseinrichtung empfangenen Schreibanforderungen; das Empfangen von Schreibanforderungen bei einer zweiten Speicherungseinrichtung, die einen assoziierten Controller, flüchtige Speicherung und nichtflüchtige Speicherung enthält; und das Verarbeiten der bei der zweiten Speicherungseinrichtung empfangenen Schreibanforderungen. Nachdem er festgestellt hat, dass die zweite Speicherungseinrichtung kurz davor ist, in eine Periode einzutreten, in der die zweite Speicherungseinrichtung nicht in der Lage sein wird, Schreibanforderungen zu verarbeiten, sendet der Controller der ersten Speicherungseinrichtung eine Commit-Synchronisierungs-Nachricht an die zweite Speicherungseinrichtung, zusammen mit Informationen, die eine Schreibanforderung bezeichnen, und der Controller der zweiten Speicherungseinrichtung bestätigt nach dem Empfangen der Commit-Synchronisierungs-Nachricht, dass mit der bezeichneten Schreibanforderung assoziierte Daten in die nichtflüchtige Speicherung der zweiten Speicherungseinrichtung geschrieben wurden. Nach dem Senden der Commit-Synchronisierungs-Nachricht akkumuliert

der Controller der ersten Speicherungseinrichtung von neuen Schreibanforderungen betroffene Speicherungsgebiete in einer Bitmap. Wenn die zweite Speicherungseinrichtung wieder in der Lage ist, Schreibanforderungen zu verarbeiten, verwendet der Controller der ersten Speicherungseinrichtung die Bitmap, um die Speicherungsregionen der ersten Speicherungseinrichtung zu identifizieren, die von der ersten Speicherungseinrichtung zur zweiten Speicherungseinrichtung kopiert werden sollen und kopiert den Inhalt der identifizierten Gebiete der ersten Speicherungseinrichtung zur zweiten Speicherungseinrichtung.

[0015] Umsetzungen der vorangehend erörterten Techniken können ein Verfahren oder einen Prozess, eine Vorrichtung oder ein System oder Computersoftware auf einem computerzugänglichen Medium umfassen.

[0016] Die Einzelheiten der einen oder mehreren Umsetzungen sind in den beiliegenden Zeichnungen und in der nachfolgenden Beschreibung aufgeführt. Weitere Merkmale und Vorteile werden aus den Beschreibungen und Zeichnungen und aus den Patentsprüchen offensichtlich.

BESCHREIBUNG DER ZEICHNUNGEN

[0017] [Fig. 1](#) ist ein Blockdiagramm eines gespiegelten Festplattensystems.

[0018] [Fig. 2](#) ist ein Ablaufdiagramm eines Verfahrens zum Überwachen von Unterschieden zwischen gespiegelten Platten.

[0019] [Fig. 3](#) ist ein Ablaufdiagramm eines Verfahrens zum Retten der Synchronisierung eines gespiegelten Plattensatzes, der abweichend geworden ist.

[0020] [Fig. 4](#) ist ein Ablaufdiagramm, das ein periodisches Synchronisierungsverfahren veranschaulicht, wie es von einem Master-Ein-/Ausgabe-Controller ausgeführt wird.

[0021] [Fig. 5](#) ist ein Ablaufdiagramm, das ein periodisches Synchronisierungsverfahren veranschaulicht, wie es von einem Slave-Ein-/Ausgabe-Controller ausgeführt wird.

[0022] [Fig. 6](#) ist ein Ablaufdiagramm zum Wiederherstellen einer gespiegelten Platte mit abweichenden Daten zu einer gespiegelten Kopie mit identischen Daten.

[0023] [Fig. 7](#) ist ein Ablaufdiagramm, das eine periodische Synchronisierung veranschaulicht, die während eines Rettungsverfahrens ausgeführt wird.

[0024] Gleiche Bezugszeichen in den verschiede-

nen Zeichnungen kennzeichnen gleiche Elemente.

AUSFÜHRLICHE BESCHREIBUNG

[0025] [Fig. 1](#) zeigt ein Blockdiagramm eines Spiegelsatzes **100**, der eine erste Datenspeicherungseinrichtung **105** und eine zweiten Datenspeicherungseinrichtung **110** umfasst. In der Umsetzung von [Fig. 1](#) handelt es sich bei den Datenspeicherungseinrichtungen um Plattenlaufwerke. In anderen Umsetzungen kann es sich bei den Datenspeicherungseinrichtungen um Gruppen von Plattenlaufwerken oder andere Speicherungseinrichtungen handeln.

[0026] Um die Beschreibung zu erleichtern, wird eine der Platten als Master-Platte bezeichnet und dient als primäre Datenspeicherungseinrichtung, während die andere Platte als Slave-Platte bezeichnet wird und als redundante Sicherung dient. Wenn beide Platten aktiv sind und die selben Daten enthalten, kann der Master- bzw. Slave-Status den beiden Platten zufällig zugewiesen werden. Zu Zwecken der nachfolgend beschriebenen Synchronisierungsverfahren unterhalten Umsetzungen mit zwei Platten tatsächlich zwei Master-Slave-Beziehungen, wobei jede Platte in einer Beziehung als Master dient und in der anderen als Slave. In [Fig. 1](#) ist die Platte **105** als Master-Platte bezeichnet, während die Platte **110** als Slave-Platte bezeichnet ist.

[0027] Ein erster E/A-("Ein-/Ausgabe") Controller **115** ist mit der ersten Platte **105** assoziiert und ein zweiter E/A-Controller **120** ist mit der zweiten Platte **110** assoziiert. Die E/A-Controller **115**, **120** steuern das Lesen und Schreiben von Daten auf den Platten.

[0028] Ein Client **125**, bei dem es sich beispielsweise um einen Prozessor handeln kann, sendet die selben Schreibanforderungen **130** an beide E/A-Controller. Jede Schreibanforderung enthält Daten. Darüber hinaus ist mit jeder Schreibanforderung eine Referenzmarke, wie beispielsweise eine sequenzielle Referenznummer, assoziiert. Die E/A-Controller schreiben die Daten von den Schreibanforderungen auf ihre jeweilige Platte, so dass unter normalen Bedingungen beide Platten identische Daten enthalten. Typischerweise verarbeitet jeder E/A-Controller die Schreibanforderungen in der selben Reihenfolge. Um dies zu erreichen, verarbeiten die E/A-Controller die Schreibanforderungen in der Reihenfolge der Referenzmarken, was bedeutet, dass die E/A-Controller die Schreibanforderungen nicht in der selben Reihenfolge empfangen müssen.

[0029] Der Client **125** sendet außerdem Leseanforderungen **135** an die E/A-Controller. In einer Umsetzung reagiert, wenn beide Platten die selben Daten enthalten, nur die Master-Platte auf die Leseanforderungen **135**. In anderen Umsetzungen können die Slave-Platte oder beide Platten reagieren. Wenn die

Master-Platte ausfällt oder unzugänglich wird, wird die Slave-Platte neu als Master-Platte bezeichnet und liefert weiterhin Daten an den Client **125**. Wenn also die Platte **105** ausfallen würde, würde die Platte **110** zur Master-Platte.

[0030] Eine Platte im Spiegelsatz **100** enthält von denen ihres Peers abweichende Daten, wenn die Platte während einer gewissen Zeitperiode nicht in der Lage ist, Schreibforderungen zu verarbeiten. Wenn beispielsweise die Slave-Platte für eine Zeitperiode außer Betrieb wäre, würden sich die Daten der Slave-Platte von den Daten der Master-Platte unterscheiden. Wenn die Platten in einem Spiegelsatz abweichend werden, kann eine Spiegelsatzkopie implementiert werden, um Daten von der Platte mit den "guten" Daten zu der Platte mit abweichenden Daten zu kopieren. Bei einigen größeren Speichereinrichtungen kann dieses Verfahren eine lange Zeit in Anspruch nehmen, während der das Niveau der Fehlertoleranz des Systems reduziert ist, da die gespiegelten Platten nicht identische Daten enthalten.

[0031] Um das Niveau der Fehlertoleranz eines Systems zu verbessern, kann die Länge der benötigten Zeit zum Wiederherstellen der gespiegelten Platten zu einem Zustand, in dem beide Platten identische Daten enthalten (was als Rettung bezeichnet werden kann), verkürzt werden, indem von der Platte mit "guten" Daten nur Abschnitte von Daten kopiert werden, die nicht auf der Platte mit abweichenden Daten gespeichert wurden. Dieses Verfahren des Kopierens von nur Abschnitten der Platte kann als Inkremental-Divergenz-Kopieren oder Delta-Kopieren bezeichnet werden (wobei sich Delta auf die Änderungen bezieht, die an einer Platte vorgenommen wurden und an einer anderen Platte nicht).

[0032] Im Allgemeinen kann Inkremental-Divergenz-Kopieren erreicht werden, indem Slave-Änderungen an einer oder mehreren gespiegelten Platten vorgenommen werden, so dass nach einer Periode der Nichtverfügbarkeit eine gespiegelte Platte mit abweichenden Daten wiederhergestellt werden kann, indem von der Platte mit "guten" Daten nur die Daten kopiert werden, die nicht auf der gespiegelten Platte mit abweichenden Daten gespeichert wurden. Das Überwachen von Änderungen, die an einer gespiegelten Platte vorgenommen wurde, erfordert allgemein das Verfolgen der Änderungen, die nach einem Zeitpunkt erfolgt sind, an dem bekannt ist, dass beide gespiegelten Platten im gespiegelten Satz identische Daten enthalten, wobei zu diesem Zeitpunkt die gespiegelten Platten als synchronisiert bezeichnet werden können.

[0033] Die Überwachung von Änderungen, die nach einem Punkt der Synchronisierung an einer gespiegelten Platte vorgenommen wurden, kann problematisch sein, wenn ein System, Subsystem oder Pro-

zessor eine Schreibforderung als abgeschlossen aufzeichnet, wenn die Daten in einen flüchtigen Platten-Cache geschrieben wurde, aber die Daten noch nicht in die nichtflüchtige Speicherung der gespiegelten Platte geschrieben wurden. Dieses Problem kann von besonderer Bedeutung sein, wenn die gespiegelten Daten in Streifen auf mehr als einer Platte angeordnet sind, beispielsweise unter Verwendung von RAID-("Redundant Array of Inexpensive Disks") Techniken, da die Zeitperiode ab der Platzierung der Schreibforderung in den Platten-Cache bis dann, wenn alle Daten in die nichtflüchtige Plattenspeicherung geschrieben wurden, infolge der verlängerten Zeit, die zum Schreiben auf mehr als eine RAID-Platte erforderlich ist, erheblich sein kann.

[0034] Die Wirksamkeit des Inkremental-Divergenz-Kopierens kann verbessert werden, indem die Daten auf den Spiegelplatten periodisch bis zu einer bestimmten Schreibforderungs-Referenzmarke synchronisiert werden (z. B. wo jede Platte identische Daten enthält), indem der Platten-Cache geleert wird und die Daten im Cache in der Plattenspeicherung festgeschrieben werden. Das Leeren des Platten-Caches stellt sicher, dass alle verarbeiteten Schreibforderungen in nichtflüchtiger Plattenspeicherung gespeichert wurden.

[0035] Jeder E/A-Controller **115**, **120** im Spiegelsatz **100** verfolgt die Schreibforderungen **130**, die an die jeweilige Platte des E/A-Controllers gemacht werden, indem er die an der Platte vorgenommenen Änderungen in einer Bitmap **155**, **156** akkumuliert. Die Bitmap **155**, **156** ist eine Datenstruktur, die eines oder mehrere Bits verwendet, um anzuzeigen, ob jedes Gebiet einer Platte von einer Schreibforderung **130** betroffen wurde. Die Bitmap in dieser Umsetzung wird auf der Platte gespeichert. Andere Umsetzungen können die Bitmap in flüchtigem Speicher speichern, bis das System abgeschaltet wird oder können die Bitmap in nichtflüchtiger Speicherung speichern, die nicht im Spiegelsatz enthalten ist. Der von der Bitmap vorgesehene Grad der Abstraktion (oder die Granularität) beruht auf der Größe des von einem Bit repräsentierten Speichergebiets. Jedes Bit repräsentiert typischerweise wesentlich mehr Daten als von einer einzigen Schreibforderung in ein entsprechendes Speicherungsgebiet geschrieben werden. Hier kann die Bitmap **155**, **156** als Plattenänderungs-Bitmap bezeichnet werden.

[0036] Die Bitmap und die Platte können durch einen eindeutigen Identifikator assoziiert sein. Der eindeutige Identifikator kann beispielsweise einen Plattenidentifikator enthalten, der die Instanz des Client **125** identifiziert, für den die Bitmap und die Platte gelten. Die Assoziation der Bitmap und der Platte kann sicherstellen, dass die veränderten Plattengebiete zur korrekten Platte kopiert werden. Beispielsweise ist die Assoziation einer bestimmten Bitmap mit einer

bestimmten Platte oder einem bestimmten Datensatz auf einer Platte wichtig, wenn der Spiegelsatz eine entnehmbare Platte enthält (d. h. eine Platte, die herausgenommen werden kann, ohne die Computergehäuseeinheit zu öffnen).

[0037] Ein E/A-Controller, der als Master-E/A-Controller bezeichnet werden kann sendet periodisch eine Commit-Synchronisierungs-Nachricht **160** an den anderen E/A-Controller, der als Slave-E/A-Controller bezeichnet werden kann. Wie gezeigt und nachfolgend beschrieben, ist der erste E/A-Controller **115** der Master-E/A-Controller und der zweite E/A-Controller **120** ist der Slave-E/A-Controller. Es ist jedoch wichtig, zu beachten, dass in der Beziehung, in der die Platte **110** der Master und die Platte **105** der Slave ist, der zweite E/A-Controller **120** gleichzeitig als Master-E/A-Controller dient (und der erste E/A-Controller gleichzeitig als Slave-E/A-Controller dient).

[0038] Die Commit-Synchronisierungs-Nachricht **160** identifiziert eine Schreibanforderungs-Referenzmarke, bis zu der die Daten auf den gespiegelten Platten synchronisiert werden sollen. Der erste E/A-Controller **115** erstellt eine Sicherungskopie **165** der Plattenänderungs-Bitmap **155**, um die Rettung zu ermöglichen, falls während des Synchronisierungsverfahrens ein Ausfall stattfindet und beginnt eine neue Plattenänderungs-Bitmap für die Verwendung im nächsten Synchronisierungsschritt, um alle nachfolgenden Schreibvorgänge zu akkumulieren.

[0039] Wenn der zweite E/A-Controller **120** die vom ersten E/A-Controller **115** gesendete Commit-Synchronisierungs-Nachricht empfängt, bestimmt der zweite E/A-Controller **120**, ob der zweite E/A-Controller die in der Commit-Synchronisierungs-Nachricht identifizierte Schreibanforderung und alle vorausgegangenen Schreibanforderungen bereits verarbeitet hat. Falls nicht, wartet der zweite E/A-Controller, bis er diese Schreibanforderung und alle vorausgegangenen Schreibanforderungen verarbeitet hat, bevor er die Synchronisierung auslöst.

[0040] Sobald der zweite E/A-Controller **120** die in der Commit-Synchronisierungs-Nachricht identifizierte Schreibanforderung und alle vorausgegangenen Schreibanforderungen verarbeitet hat, oder falls der zweite E/A-Controller die Schreibanforderung und alle vorausgegangenen Schreibanforderungen bereits verarbeitet hatte, als die Commit-Synchronisierungs-Nachricht empfangen wurde, leert der zweite E/A-Controller seinen Platten-Controller-Cache, um die verarbeiteten Schreibanforderungen in nichtflüchtiger Plattenspeicherung festzuschreiben. Wenn die Cache-Leerung erfolgreich ist, sendet der zweite E/A-Controller **120** eine Bestätigungsnachricht **170** an den ersten E/A-Controller **115**. Wenn er die Bestätigung empfängt, dass Leerung und Synchronisierung erfolgreich waren, löscht der erste E/A-Controll-

er **115** die Sicherungskopie **165** seiner Plattenänderungs-Bitmap. Falls die Synchronisierung misslingt oder falls der erste E/A-Controller **115** innerhalb einer vorherbestimmten Zeit keine Bestätigung empfängt, kombiniert der erste E/A-Controller **115** die Bitmap **155** und die Sicherung **165** (typischerweise durch ODER-Verknüpfung) und verwendet die kombinierte Bitmap bei Wiederherstellen der zweiten Platte **110**.

[0041] Ein Inkremental-Divergenz-Kopierverfahren, das nur von einem bestimmten Punkt an einer Platte vorgenommene Änderungen akkumuliert, kann ausgelöst werden, wenn ein Plattenausfall erkannt wird (und akkumuliert so nur Änderungen, die während einer Periode der Nichtverfügbarkeit vorgenommen werden) oder kann verwendet werden, wann immer das System aktiv ist (und akkumuliert so Änderungen, die zu jeder Zeit während des Systemsbetriebs am Spiegelsatz vorgenommen werden). Wenn Änderungen nur während der Periode der Nichtverfügbarkeit akkumuliert werden, muss die Periode der Nichtverfügbarkeit mit einer Platten-Controller-Cache-Leerung für die Platte beginnen, die unverfügbar wird (was durch ein Verfahren geschehen kann, wenn die Platte unverfügbar wird, das als "sanfte Abschaltung" bezeichnet wird), damit das Inkremental-Divergenz-Kopierverfahren wirksam die Platte wiederherstellen kann, nachdem sie verfügbar wird.

[0042] Bei einer weiteren Umsetzung können Änderungen durch Löschen bestimmter Schreibanforderungen in der Plattenänderungs-Bitmap akkumuliert werden, statt dass begonnen wird, Plattenänderungen zu akkumulieren, die nach einem Synchronisierungspunkt in einer anderen Plattenänderungs-Bitmap vorgenommen wurden. Dadurch kann die Rettungszeit verkürzt werden.

[0043] Beide E/A-Controller **115**, **120** im Spiegelsatz **100** können das Commit-Synchronisierungs-Verfahren auslösen, um sicherzustellen, dass bis zu einer bestimmten Schreibanforderungs-Referenzmarke beide Platten die selben Daten enthalten. Nach einer Periode der Nichtverfügbarkeit kann die gespiegelte Platte mit abweichenden Daten zu einer Spiegelkopie, die identische Daten speichert, wiederhergestellt werden, indem nur die Plattenregionen kopiert werden, die seit der letzten Synchronisierung geändert wurden.

[0044] Bei einer Umsetzung kann ein Datum (oder ein Datum und eine Uhrzeit) mit einem bestimmten Datum, das auf einer der gespiegelten Platten gesetzt ist, assoziiert sein. Das kann von Vorteil sein, wenn Schreibanforderungs-Referenzmarken nicht unbedingt eindeutig sind. Beispielsweise ist eine Schreibanforderungs-Referenzmarke möglicherweise nicht eindeutig, wenn es sich bei der Referenzmarke um eine sequenzielle Nummer handelt, die bei irgend einem festen Wert neu beginnt (z. B. eins),

wenn das den Client steuernde Betriebssystem zurückgesetzt wird. Solche Schreibanforderungen können eindeutig identifiziert werden, indem ein Datum (oder ein Datum und eine Uhrzeit) mit dem Datensatz assoziiert wird, das gesetzt wird, wenn der Platten-Cache der den Datensatz speichernden Platte geleert wird. Wenn der Client neu gestartet wird, kann alternativ oder zusätzlich einem Spiegelsatz eine neue Instanznummer gegeben werden, um die Unterscheidung nicht eindeutiger Schreibanforderungs-Referenzmarken zu erleichtern. Andere eindeutige Identifikatoren können, einzeln oder in Kombination, einen Client-Identifikator, einen Spiegelsatz-identifikator und einen Datensatz-identifikator umfassen.

[0045] Obwohl [Fig. 1](#) für veranschaulichende Zwecke zwei Platten als Spiegeleinrichtungen zum Speichern der gespiegelten Datensätze nutzt, sind die Vorteile des Inkremental-Divergenz-Kopierens nicht auf diese bestimmte Umsetzung beschränkt und sind ebenso auf andere Umsetzungen mit anderen Zahlen oder Typen von Speichervorrichtung, einschließlich RAID-Technik anwendbar. Beispielsweise können andere Umsetzungen drei oder mehr Platten spiegeln oder können mehrere Exemplifizierungen einer gespiegelten Platte vorsehen (z. B. können vier Platten verwendet werden, um zwei gespiegelte Sätze für die selbe Platte vorzusehen).

[0046] Wie in [Fig. 2](#) zu sehen, verwendet ein Verfahren **200** Inkremental-Divergenz-Verfolgung, um die Wiederherstellung der Synchronisierung für einen gespiegelten Plattensatz vorzubereiten, der während einer Periode, in der eine Platte des gespiegelten Plattensatzes durch eine sanfte Abschaltung unverfügbar wurde, abweichend geworden ist. Die Umsetzung eines gespiegelten Plattensatzes in [Fig. 2](#) hat zwei Plattenspeicherungseinrichtungen, die jeweils durch einen getrennten E/A-Controller gesteuert werden. Jeder E/A-Controller empfängt die selben Schreibanforderungen von einem Prozessor und verarbeitet die empfangenen Schreibanforderungen in sequenzieller Reihenfolge. Mit jeder Schreibanforderung ist eine Referenzmarke assoziiert, die beim Sequenzialisieren der Schreibanforderungen verwendet wird. Eine weitere Umsetzung kann verfolgen, ob eine bestimmte Schreibanforderung verarbeitet (oder abgeschlossen) wurde oder nicht. Dadurch wird zugelassen, dass Anforderungen in beliebiger Reihenfolge (d. h. nicht sequenziell) verarbeitet werden. Wenn beide Platten aktiv sind, enthalten die Platten jeweils die selben Daten.

[0047] Das Verfahren **200** wird ausgelöst, wenn eine Feststellung erfolgt, dass eine der Platten in eine Periode der Nichtverfügbarkeit eintreten wird (Schritt **205**). Wenn diese Feststellung erfolgt, wird der E/A-Controller für die Platte, die unverfügbar wird, angewiesen, Schreibanforderungen, die vom

E/A-Controller verarbeitet wurden, in nichtflüchtiger Speicherung festzuschreiben (Schritt **210**). Die unverfügbar werdende Platte kann als Slave-Platte bezeichnet werden und die aktive Platte kann als Master-Platte bezeichnet werden und die assoziierten E/A-Controller können als Slave-E/A-Controller und Master-E/A-Controller bezeichnet werden. Der Master-E/A-Controller beginnt, an der Master-Platte vorgenommene Änderungen in einer Plattenänderungs-Bitmap zu akkumulieren (Schritt **220**), empfängt und verarbeitet weiterhin Schreibanforderungen (Schritt **225**) und aktualisiert die Plattenänderungs-Bitmap, um jede aus den verarbeiteten Schreibanforderungen resultierende Änderung an der Master-Platte zu reflektieren (Schritt **230**). Jedes Bit in der Plattenänderungs-Bitmap repräsentiert ein Gebiet der Master-Platte. Andere Umsetzungen können die Menge des von jedem Bit in der Plattenänderungs-Bitmap repräsentierten Speicherplatzes auf der Platte variieren.

[0048] Der Master-E/A-Controller überwacht außerdem weiterhin den Status der Slave-Platte (Schritt **235**). Wenn die Slave-Platte verfügbar wird und begonnen hat, neue Schreibanforderungen zu verarbeiten, beginnt der Master-E/A-Controller ein Rettungs-verfahren **300**, wie nachfolgend anhand von [Fig. 3](#) beschrieben (Schritt **240**).

[0049] Wie in [Fig. 3](#) zu sehen, werden im Rettungs-verfahren **300** Abschnitte der Master-Platte gemäß Angabe in der Plattenänderungs-Bitmap zur Slave-Platte kopiert. Das Rettungs-verfahren findet als Hintergrundverfahren statt, das aktiv ist, während der Spiegelsatz weiterhin neue Schreibanforderungen verarbeitet. Das Rettungs-verfahren **300** beginnt, wenn der Master-E/A-Controller eine Sicherungskopie der Plattenänderungs-Bitmap macht und die Originalversion der Plattenänderungs-Bitmap als Rettungs-Bitmap bezeichnet (Schritt **310**). Der Master-E/A-Controller beginnt außerdem eine neue Plattenänderungs-Bitmap, um alle nachfolgenden Änderungen der Master-Platte zu akkumulieren (Schritt **320**). Die Sicherungskopie der Plattenänderungs-Bitmap und die neue Plattenänderungs-Bitmap ermöglichen die Rettung, falls ein Ausfall während des Rettungs-verfahrens stattfindet.

[0050] Der Master-E/A-Controller prüft jedes Bit in der Rettungs-Bitmap (Schritt **330**) und stellt fest, ob das Bit anzeigt, dass das entsprechende Gebiet der Master-Platte verändert wurde (Schritt **340**). Falls nicht, prüft der Master-E/A-Controller dann das nächste Bit in der Rettungs-Bitmap (Schritt **345**). Wenn das Bit anzeigt, dass das Gebiet der Master-Platte verändert wurde, bestimmt der Master-E/A-Controller, ob nachfolgende Schreibanforderungen das entsprechende Gebiet der Slave-Platte verändert haben (Schritt **345**).

[0051] Wenn nachfolgende Schreibanforderungen das entsprechende Gebiet der Slave-Platte verändert haben, kopiert der Master-E/A-Controller nur den Abschnitt von dem Gebiet der Master-Platte, der dem Abschnitt des Gebiets der Slave-Platte entspricht, der nicht durch eine nachfolgende Schreibanforderung verändert wurde (Schritt **350**). Der Master-E/A-Controller kann den Abschnitt identifizieren, der nicht verändert wurde, indem er den Slave-E/A-Controller eine Liste von Schreibanforderungen führen lässt, die während des Rettungsverfahrens von der Slave-Platte verarbeitet wurden, wobei jeder Eintrag in der Liste den tatsächlichen Speicherabschnitt identifiziert, der verändert wurde. Alternativ kann der Slave-E/A-Controller eine Plattenänderungs-Bitmap mit feinerer Granularität unterhalten, so dass jedes Bit der Bitmap dem kleinsten Abschnitt der Platte entspricht, den eine Schreibanforderung verändern darf. Um Platz zu sparen, kann der Slave-E/A-Controller eine Bitmap mit veränderlicher Granularität unterhalten, so dass eine Bitmap mit feinerer Granularität nur für veränderte Abschnitte der Platte unterhalten wird.

[0052] Wenn keine nachfolgenden Änderungen an dem Gebiet der Slave-Platte vorgenommen wurden, kopiert der Master-E/A-Controller das gesamte Gebiet der Master-Platte zur Slave-Platte (Schritt **355**).

[0053] Der Master-E/A-Controller verändert den Abschnitt der Daten, der kopiert wird, um eine potenzielle Ineffizienz beim Schreiben von Daten zu vermeiden, die von einer nachfolgenden Schreibanforderung überschrieben werden (Schritte **345–355**). Wenn beispielsweise eine Schreibanforderung WR-102 einen Abschnitt der im Plattengebiet **12** gespeicherten Daten verändert und eine Schreibanforderung WR-155 außerdem in einem anderen Abschnitt des Plattengebiets **12** gespeicherte Daten verändert, kann das Verfahren zum Schreiben von Daten in das Plattengebiet **12** nur die Abschnitte der Region **12** verändern, die für jede Schreibanforderung benötigt werden.

[0054] Außerdem oder alternativ kann der Slave-E/A-Controller den Abschnitt der Daten verändern, der kopiert wird. Wenn beispielsweise der Slave-E/A-Controller eine neue Schreibanforderung empfangen hat, die das selbe Plattengebiet verändert, das durch die von der Master-Platte kopierten Daten aktualisiert werden soll, kann der Slave-E/A-Controller den Abschnitt der Daten verändern, der von der Master-Platte kopiert wird.

[0055] Nach dem Kopieren des Gebiets der Master-Platte (bzw. eines Abschnitts desselben) zur Slave-Platte bestimmt der Master-E/A-Controller, ob mehr Bits in der Rettungs-Bitmap geprüft werden müssen (Schritt **360**) und falls ja, prüft er das nächste Bit (Schritt **330**).

[0056] Die Rettung ist abgeschlossen, wenn der Master-E/A-Controller feststellt, dass alle Bits in der Rettungs-Bitmap geprüft wurden. Nach dem Abschluss kann der Master-E/A-Controller optional eine Synchronisierung auslösen, die den Slave-Platten-Cache leert (Schritt **370**), um die kopierten Daten auf der Slave-Platte festzuschreiben. Wenn der Master E/A-Controller feststellt, dass eine nachfolgende Synchronisierung oder Leerung nicht erfolgreich ist (Schritt **375**), kombiniert der Master-E/A-Controller die Sicherungskopie der Plattenänderungs-Bitmap mit der neuen Plattenänderungs-Bitmap (typischerweise durch ODER-Verknüpfung) (Schritt **380**) und wiederholt das Rettungsverfahren **300** unter Verwendung der kombinierten Plattenänderungs-Bitmap. Wenn die Slave-E/A-Controller-Synchronisierung und Slave-Platten-Controller-Leerung erfolgreich sind, löscht der Master-E/A-Controller die Sicherungsplattenänderungs-Bitmap (Schritt **390**).

[0057] Obwohl die unter Verweis auf [Fig. 2](#) erörterte Umsetzung den Grad der Granularität beim Kopieren von Plattengebieten von der Master-Platte zur Slave-Platte verändert, kann eine andere Umsetzung jedes Mal das gesamte veränderte Gebiet kopieren, ungeachtet dessen, ob ein Abschnitt des Gebiets von einer nachfolgenden Schreibanforderung verändert werden wird. Die Umsetzung von [Fig. 3](#) verarbeitet die Rettungs-Bitmap während der Rettung, so dass sie kein zweites Mal verwendet werden kann. Um eine Rettung bei einem Ausfall während des Rettungsverfahrens zu ermöglichen, wird vor dem Verarbeiten der Bits eine Sicherungskopie der Plattenänderungs-Bitmap erstellt (Schritt **310**). Eine weitere Umsetzung zerstört die Rettungs-Bitmap während der Rettung nicht und kann von einem Ausfall während des Rettungsverfahrens gerettet werden, indem die Rettungs-Bitmap selbst verwendet wird. Diese Umsetzung erstellt vor dem Verarbeiten von Bits keine Kopie der Plattenänderungs-Bitmap (Schritt **310**). Eine alternative Umsetzung verwendet keine neue Plattenänderungs-Bitmap zum Akkumulieren von Master-Plattenänderungen, die vorgenommen wurden, nachdem die Slave-Platte zur Verfügbarkeit zurückgekehrt ist, aber bevor das Rettungsverfahren erfolgreich abgeschlossen wurde.

[0058] Wie in [Fig. 4–Fig. 6](#) zu sehen, kann ein Inkremental-Divergenz-Kopierverfahren jedes Mal aktiv sein, wenn der Spiegelsatz genutzt wird. Die Umsetzung eines gespiegelten Satzes in [Fig. 4–Fig. 6](#) hat zwei Plattenspeicherungseinrichtungen und zwei E/A-Controller, die Schreibanforderungen in der anhand von [Fig. 2](#) beschriebenen Weise empfangen.

[0059] Jeder E/A-Controller verfolgt die an die Platte des E/A-Controllers gestellten Schreibanforderungen durch Akkumulieren der an der Platte vorgenommen Änderungen in einer Plattenänderungs-Bitmap. Ein E/A-Controller (der Master-E/A-Controller genannt

wird) sendet periodisch eine Commit-Synchronisierungs-Nachricht an den anderen E/A-Controller (der Slave-E/A-Controller genannt wird), um ein periodisches Synchronisierungsverfahren zu starten. [Fig. 4](#) zeigt ein periodisches Synchronisierungsverfahren, das von einem Master-E/A-Controller ausgeführt wird. [Fig. 5](#) zeigt ein periodisches Synchronisierungsverfahren, das von einem Slave-E/A-Controller ausgeführt wird. [Fig. 6](#) zeigt ein Verfahren zum Wiederherstellen einer gespiegelten Platte mit abweichenden Daten zu einer Spiegelkopie mit identischen Daten.

[0060] Wie in [Fig. 4](#) zu sehen, löst ein Master-E/A-Controller ein Verfahren **400** aus, um eine periodisch Synchronisierung mit einem Slave-E/A-Controller auszuführen. Das Verfahren **400** beginnt, wenn ein Master-E/A-Controller Schreibanforderungen vom Prozessor empfängt und verarbeitet (Schritt **410**) und an der Master-Platte vorgenommene Änderungen in einer Plattenänderungs-Bitmap akkumuliert (Schritt **415**). Der Master-E/A-Controller stellt fest, ob der Spiegelsatz synchronisiert werden soll oder nicht (Schritt **420**). Der Master-E/A-Controller kann eine Commit-Synchronisierungs-Anforderung stellen, beispielsweise nachdem eine vorgegebene Zeitperiode seit der letzten Synchronisierung verstrichen ist, nachdem eine vorgegebene Zahl von Schreibanforderungen seit der letzten Synchronisierung verarbeitet wurden oder nach einem festen Prozentsatz inkrementeller Divergenz zwischen zwei gespiegelten Platten. Bei der Bestimmung, wann eine Synchronisierung angefordert wird, kann die Häufigkeit der Synchronisierung (die die Systemleistung verringern kann, da das Leeren des Platten-Cache die Verarbeitung aller Schreibanforderungen während der Zeit anhält, in der der Platten-Cache in nichtflüchtige Speicherung geschrieben wird) gegen die Menge von Daten abgewogen werden, die zwischen den gespiegelten Platten nicht synchronisiert sind (was eine längere Zeit zum Ausführen des inkremental-divergenten Kopierens zum Wiederherstellen identischer Daten für den Spiegelsatz erfordern kann).

[0061] Wenn der Master-E/A-Controller feststellt, dass der Spiegelsatz synchronisiert werden sollte, sendet der Master-E/A-Controller eine Commit-Synchronisierungs-Nachricht an den Slave-E/A-Controller (Schritt **430**). Die Commit-Synchronisierungs-Nachricht identifiziert eine Schreibanforderungs-Referenzmarke, bis zu der die Daten auf den gespiegelten Platten synchronisiert werden sollen. Der Master-E/A-Controller erstellt eine Sicherungskopie der Plattenänderungs-Bitmap (Schritt **435**), um die Rettung zu ermöglichen, falls während des Synchronisierungsverfahrens ein Ausfall stattfindet und beginnt eine neue Plattenänderungs-Bitmap für die Verwendung im nächsten Synchronisierungsschritt, um Plattenänderungen zu akkumulieren, die ab die-

sem Punkt an der Platte des Master-E/A-Controllers vorgenommen wurden (Schritt **440**). Der Master-E/A-Controller empfängt und verarbeitet weiterhin Schreibanforderungen vom Prozessor (Schritt **445**) und aktualisiert die neue Plattenänderungs-Bitmap, um jede Änderung der Master-Platte zu reflektieren (Schritt **450**).

[0062] Bei Empfang einer Bestätigung, dass die Cache-Leerung durch den Slave-E/A-Controller und die Synchronisierung erfolgreich waren (Schritt **455**), löscht der Master-B/A-Controller die Sicherungs-Plattenänderungs-Bitmap (Schritt **460**) und das Synchronisierungsverfahren endet.

[0063] Alternativ kann der Master E/A-Controller feststellen, dass die Synchronisierung misslungen ist (Schritt **455**), da beispielsweise der Master-E/A-Controller innerhalb einer vorherbestimmten Zeit keine Bestätigungsnachricht vom Slave-E/A-Controller empfangen hat oder der Master-E/A-Controller eine Nachricht empfangen hat, dass die Synchronisierung misslungen ist. In diesem Fall kombiniert der Master-E/A-Controller die Sicherungs-Plattenänderungs-Bitmap und die neue Plattenänderungs-Bitmap (typischerweise durch ODER-Verknüpfung) (Schritt **470**) und löst, wenn er feststellt, dass der Slave-E/A-Controller und dessen assoziierte Platte funktionsfähig sind, das unter Verweis auf [Fig. 3](#) beschriebene Rettungsverfahren **300** aus, unter Verwendung der kombinierten Plattenänderungs-Bitmap zum Angeben, welche Plattengebiete von der Master-Platte zur Slave-Platte kopiert werden sollen (Schritt **475**).

[0064] Wie in [Fig. 5](#) zu sehen, beginnt ein Verfahren **500**, wenn ein Slave-E/A-Controller eine Commit-Synchronisierungs-Nachricht empfängt, die eine Schreibanforderungs-Referenzmarke identifiziert, bis zu der die Daten auf den gespiegelten Platten zu synchronisieren sind (Schritt **510**). Der Slave-E/A-Controller bestimmt, ob der Slave-E/A-Controller die in der Commit-Synchronisierungs-Nachricht identifizierte Schreibanforderung und alle vorausgegangenen Schreibanforderungen bereits verarbeitet hat (Schritt **520**). Falls nicht, wartet der Slave-E/A-Controller, bis er diese Schreibanforderung und alle vorausgegangenen Schreibanforderungen verarbeitet hat, bevor er die Synchronisierung auslöst.

[0065] Sobald der Slave-E/A-Controller die in der Commit-Synchronisierungs-Nachricht identifizierte Schreibanforderung und alle vorausgegangenen Schreibanforderungen verarbeitet hat, oder falls der Slave-E/A-Controller die Schreibanforderung und alle vorausgegangenen Schreibanforderungen bereits verarbeitet hatte, als die Commit-Synchronisierungs-Nachricht empfangen wurde, leert der Slave-Platten-Controller seinen Cache, um die verarbeiteten Schreibanforderungen in nichtflüchtiger Plat-

tenspeicherung festzuschreiben (Schritt **530**) und bestimmt, ob die Cache-Leerung erfolgreich war (Schritt **540**). Wenn die Cache-Leerung erfolgreich war, sendet der Slave-E/A-Controller eine Bestätigungsnachricht an den Master-E/A-Controller (Schritt **550**). Wenn die Cache-Leerung nicht erfolgreich war, sendet der Slave-E/A-Controller eine Fehlernachricht an den Master-E/A-Controller (Schritt **560**). Nach dem Senden der entsprechenden Nachricht an den Master-E/A-Controller beendet der Slave-E/A-Controller das Verfahren **500**.

[**0066**] [Fig. 6](#) zeigt ein Verfahren **600** zum Wiederherstellen einer gespiegelten Platte mit abweichenden Daten zu einer gespiegelten Kopie, die identische Daten speichert. Die folgende Beschreibung geht davon aus, dass eine der Platten (die Slave-Platte) in einem gespiegelten Satz zuvor ausgefallen ist oder anderweitig un verfügbar geworden ist und dass eine Plattenänderungs-Bitmap existiert, die alle, seit dem letzten Mal, dass der gespiegelte Satz synchronisiert wurde, an der verbleibenden aktiven Platte (der Master-Platte) vorgenommenen Änderungen enthält. Das kann beispielsweise erreicht werden, indem die anhand von [Fig. 4–Fig. 5](#) beschriebenen Verfahren ausgeführt werden.

[**0067**] Wenn die Slave-Platte nicht verfügbar ist, empfängt und verarbeitet der Master E/A-Controller weiterhin Schreibanforderungen vom Prozessor (Schritt **610**) und akkumuliert weiterhin an der Master-Platte vorgenommene Änderungen in der Plattenänderungs-Bitmap, die die Plattenänderungen verfolgt, die seit der letzten Synchronisierung vorgenommen wurden (Schritt **620**). Wenn der Master-E/A-Controller feststellt, dass die Slave-Platte wieder funktionsfähig ist und in der Lage ist, mit der Verarbeitung von Schreibanforderungen zu beginnen (Schritt **630**), beginnt der Master-E/A-Controller ein Rettungsverfahren **300**, wie anhand von [Fig. 3](#) beschrieben, wobei die Plattenänderungs-Bitmap verwendet wird, um die Slave-Platte so wiederherzustellen, dass sie Daten enthält, die mit denen der Master-Platte identisch sind (Schritt **640**).

[**0068**] Das Inkremental-Divergenz-Kopieren, das durch Ausführen der Verfahren **400**, **500** und **600** erreicht wird, unterscheidet sich von dem, dass durch Ausführen des Verfahrens **200** erreicht wird. Insbesondere sind die Verfahren **400–600** wirksam, während eines unerwarteten Platten- oder Controller-Ausfalls in einer der Platten einen Spiegelplattensatz wiederherzustellen, da die Plattenänderungs-Bitmaps aktualisiert werden, während der Spiegelplatz aktiv ist. Das Verfahren **200** ist nur wirksam zum Wiederherstellen eines Spiegelplattensatzes, wenn ausreichend Warnung vor einer bevorstehenden Periode der Nichtverfügbarkeit einer Platte gegeben wird, so dass eine Platten-Cache-Leerung stattfinden kann und begonnen werden kann, Ände-

rungen an der verbleibenden aktiven Platte in einer Plattenänderungs-Bitmap zu akkumulieren. Da das Verfahren **200** jedoch nur zu bestimmten Zeiten umgesetzt wird, kann es zu deutlich geringeren Verarbeitungs-Gesamtkosten führen als die Verfahren **400–600**.

[**0069**] Wie in [Fig. 7](#) zu sehen, umfasst ein Rettungsverfahren **700** das Ausführen periodischer Synchronisierung und Kopierens von Abschnitten der Master-Platte gemäß Angabe durch die Plattenänderungs-Bitmap zur Slave-Platte. Das Rettungsverfahren **700** beginnt, wenn der Master-E/A-Controller eine Sicherungskopie der Plattenänderungs-Bitmap erstellt und die Originalversion der Plattenänderungs-Bitmap als Rettungs-Bitmap bezeichnet (Schritt **710**). Der Master-E/A-Controller beginnt außerdem eine neue Plattenänderungs-Bitmap, um alle anschließenden Änderungen der Master-Platte zu akkumulieren (Schritt **720**).

[**0070**] Wie vorangehend anhand von [Fig. 3](#) beschrieben, prüft der Master-E/A-Controller jedes Bit in der Rettungs-Bitmap (Schritt **730**) und wenn das Bit angibt, dass das Gebiet der Master-Platte verändert wurde, kopiert die Master-Platte die veränderten Abschnitte der Master-Platte zur Slave-Platte (Schritt **735**).

[**0071**] Wie vorangehend anhand von [Fig. 4](#) beschrieben, löst der Master-E/A-Controller periodisch ein Synchronisierungsverfahren mit dem Slave-E/A-Controller aus. Wenn insbesondere der Master-E/A-Controller feststellt, dass der Spiegelsatz synchronisiert werden sollte (Schritt **740**), sendet der Master-E/A-Controller eine Commit-Synchronisierungs-Nachricht an den Slave-E/A-Controller, erstellt eine Sicherungskopie der Plattenänderungs-Bitmap und beginnt eine neue Plattenänderungs-Bitmap, um an der Platte des Master-E/A-Controllers vorgenommene Änderungen ab diesem Punkt zu akkumulieren (Schritt **745**).

[**0072**] Bei Empfang einer Bestätigung, dass die Cache-Leerung durch den Slave-Platten-Controller und die Synchronisierung erfolgreich waren, (Schritt **750**), entfernt der Master-E/A-Controller die Bits aus der Rettungs-Bitmap, die Gebiete auf der Master-Platte anzeigen, die erfolgreich zur Slave-Platte kopiert wurden (Schritt **755**). Der Master-E/A-Controller kann dies beispielsweise erreichen, indem er eine Liste der von der Master-Platte während des Rettungsverfahrens verarbeiteten Bits führt und die aufgelisteten Bits aus der Rettungs-Bitmap löscht. Wenn der Master-E/A-Controller jedoch feststellt, dass die Synchronisierung nicht erfolgreich war (Schritt **750**), kombiniert der Master-E/A-Controller die Sicherungs-Plattenänderungs-Bitmap und die neue Plattenänderungs-Bitmap (Schritt **760**) und löst das Rettungsverfahren **300**, wie anhand [Fig. 3](#) beschrieben, unter

Verwendung der kombinierten Plattenänderungs-Bitmap aus (Schritt 765).

[0073] Wenn der Master-E/A-Controller festgestellt hat, dass alle Bits in der Rettungs-Bitmap geprüft wurden (Schritt 770), ist das Rettungsverfahren vollständig und die Sicherungs-Plattenänderungs-Bitmap wird gelöscht (Schritt 775). Umsetzungen können ein Verfahren oder einen Prozess, eine Vorrichtung oder ein System oder Computersoftware auf einem Computermedium umfassen. Es versteht sich, dass verschiedene Abwandlungen vorgenommen werden können, ohne vom Umfang der nachfolgenden Patenansprüche abzuweichen.

Patentansprüche

1. Verfahren zum Unterhalten einer gespiegelten Kopie einer ersten Speicherungseinrichtung (105) bei einer zweiten Speicherungseinrichtung (110) in einem Computersystem, wobei das Verfahren Folgendes umfasst:

Empfangen von Schreibanforderungen (130) bei einer ersten Speicherungseinrichtung (105), wobei die erste Speicherungseinrichtung (105) einen assoziierten Controller (115) enthält;

Verarbeiten der bei der ersten Speicherungseinrichtung (105) empfangenen Schreibanforderungen (130);

Empfangen von Schreibanforderungen (130) bei einer zweiten Speicherungseinrichtung (110), wobei die zweite Speicherungseinrichtung (110) einen assoziierten Controller (120), flüchtige Speicherung und nichtflüchtige Speicherung enthält;

Verarbeiten der bei der zweiten Speicherungseinrichtung (110) empfangenen Schreibanforderungen (130);

Senden einer Commit-Synchronisierungs-Nachricht (160) zusammen mit einer Schreibanforderung (130) bezeichnenden Informationen an die zweite Speicherungseinrichtung (110), wobei es sich bei den mit der Commit-Synchronisierungs-Nachricht (160) gesendeten Informationen um eine Referenzmarke handelt, die eine von der ersten Speicherungseinrichtung (105) verarbeitete Schreibanforderung identifiziert, wobei die Referenzmarken sequenziell Schreibanforderungen (130) zugeordnet sind; und

Veranlassen, dass die zweite Speicherungseinrichtung (110) nach dem Empfangen der Commit-Synchronisierungs-Nachricht (160) bestätigt, dass Daten, die mit allen Schreibanforderungen (130) assoziiert sind, die der bezeichneten Schreibanforderung vorausgegangen sind, in die nichtflüchtige Speicherung der zweiten Speicherungseinrichtung (110) geschrieben worden sind und dass Daten, die mit der bezeichneten Schreibanforderung assoziiert sind, die durch die mit der Commit-Synchronisierungs-Nachricht (160) gesendeten Informationen identifiziert ist, in die nichtflüchtige Speicherung der zweiten Speicherungseinrichtung (110) geschrieben worden sind.

2. Verfahren nach Anspruch 1, weiterhin umfassend zu veranlassen, dass die zweite Speicherungseinrichtung (110) nach dem Empfangen der Commit-Synchronisierungs-Nachricht (160) die bezeichnete Schreibanforderung (130) verarbeitet.

3. Verfahren nach Anspruch 1 oder 2, weiterhin umfassend zu veranlassen, dass die zweite Speicherungseinrichtung (110) nach dem Empfangen der Commit-Synchronisierungs-Nachricht (160) eine erfolgreiche Cache-Leerung der flüchtigen Speicherung der zweiten Speicherungseinrichtung (110) bestätigt.

4. Verfahren nach einem der vorhergehenden Ansprüche, wobei das Verarbeiten der bei der zweiten Speicherungseinrichtung (110) empfangenen Schreibanforderungen weiterhin das Verarbeiten von Schreibanforderungen (130) in sequentieller Reihenfolge nach ihren Referenzmarken derart umfasst, dass alle Schreibanforderungen (130), die vor der von der Referenzmarke in der zweiten Commit-Synchronisierungs-Nachricht (160) identifizierten Schreibanforderung ausgegeben werden, vor der Verarbeitung dieser Schreibanforderung verarbeitet werden.

5. Verfahren nach einem der vorhergehenden Ansprüche, wobei jede Speicherungseinrichtung Schreibanforderungen (130) mit der gleichen Sequenz von Referenzmarken empfängt.

6. Verfahren nach einem der vorhergehenden Ansprüche, weiterhin umfassend das Identifizieren von Speicherungsgebieten, die von Schreibanforderungen (130) betroffen sind, die bei der ersten Speicherungseinrichtung (105) verarbeitet worden sind.

7. Verfahren nach Anspruch 6, wobei das Identifizieren von Speicherungsgebieten, die von Schreibanforderungen (130) betroffen sind, die bei der ersten Speicherungseinrichtung (105) verarbeitet worden sind, weiterhin das Akkumulieren der identifizierten Speicherungsgebiete in einer ersten Bitmap (155) umfasst.

8. Verfahren nach Anspruch 7, weiterhin umfassend:

Akkumulieren von neu identifizierten Speicherungsgebieten in einer zweiten Bitmap (156) nach dem Senden der Commit-Synchronisierungs-Nachricht (160);

Senden einer Statusnachricht an die erste Speicherungseinrichtung (105), die anzeigt, ob die Schreibdaten erfolgreich in die nichtflüchtige Speicherung geschrieben wurden, nachdem die zweite Speicherungseinrichtung (110) bestätigt, dass Daten in den verarbeiteten Schreibanforderungen (130) in eine nichtflüchtige Speicherung der zweiten Speicherungseinrichtung (110) geschrieben worden sind; und

Löschen der ersten Bitmap (**155**) nach dem Empfangen der Statusnachricht, die anzeigt, dass die Schreibdaten erfolgreich geschrieben wurden.

9. Verfahren nach Anspruch 8, weiterhin umfassend das Kopieren des Inhalts der zweiten Bitmap (**156**) in die erste Bitmap (**155**) und Löschen der zweiten Bitmap (**156**) nach dem Empfangen der Statusnachricht, die anzeigt, dass die Schreibdaten nicht erfolgreich geschrieben wurden.

10. Verfahren nach Anspruch 8, weiterhin umfassend das Bezeichnen der zweiten Bitmap (**156**) als der ersten Bitmap (**155**) nach dem Löschen der ersten Bitmap (**155**).

11. Verfahren nach einem der Ansprüche 7 bis 10, weiterhin umfassend, nach einer Periode, wenn die zweite Speicherungseinrichtung (**110**) nicht in der Lage war, Schreibanforderungen (**130**) zu verarbeiten:

Kopieren des Inhalts der ersten Bitmap (**155**) in eine Wiederherstellungsbitmap,
Verwenden der Wiederherstellungsbitmap zum Identifizieren von Speicherungsgebieten der ersten Speicherungseinrichtung (**105**), die von der ersten Speicherungseinrichtung (**105**) zu der zweiten Speicherungseinrichtung (**110**) kopiert werden sollen,
Kopieren der identifizierten Speicherungsgebiete der ersten Speicherungseinrichtung (**105**) zu der zweiten Speicherungseinrichtung (**110**) und
Akkumulieren neu empfangener Schreibanforderungen (**130**) an der ersten Speicherungseinrichtung (**105**) in einer dritten Bitmap (**165**).

12. Verfahren nach einem der vorhergehenden Ansprüche, wobei die erste Speicherungseinrichtung (**105**) eine flüchtige Speicherung und eine nichtflüchtige Speicherung enthält, wobei das Verfahren weiterhin umfasst: Senden einer zweiten Commit-Synchronisierungs-Nachricht (**160**) zusammen mit Informationen, die eine zweite Schreibanforderung bezeichnen, an die erste Speicherungseinrichtung (**105**) und Veranlassen, dass die erste Speicherungseinrichtung (**105**) nach dem Empfangen der zweiten Commit-Synchronisierungs-Nachricht (**160**) bestätigt, dass mit der bezeichneten zweiten Schreibanforderung assoziierte Daten in die nichtflüchtige Speicherung der ersten Speicherungseinrichtung (**105**) geschrieben worden sind.

13. Verfahren nach Anspruch 12, wobei das Veranlassen, dass die erste Speicherungseinrichtung (**105**) bestätigt, dass mit der bezeichneten zweiten Schreibanforderung assoziierte Daten in die nichtflüchtige Speicherung der ersten Speicherungseinrichtung (**105**) geschrieben worden sind, umfasst zu veranlassen, dass die erste Speicherungseinrichtung (**105**) bestätigt, dass Daten, die mit allen Schreibanforderungen (**130**) assoziiert sind, die der bezeichne-

ten zweiten Schreibanforderung vorausgegangen sind, in die nichtflüchtige Speicherung der ersten Speicherungseinrichtung (**105**) geschrieben worden sind.

14. Verfahren nach Anspruch 13, wobei das Veranlassen, dass die erste Speicherungseinrichtung (**105**) bestätigt, dass mit der bezeichneten zweiten Schreibanforderung assoziierte Daten in die nichtflüchtige Speicherung der ersten Speicherungseinrichtung (**105**) geschrieben worden sind, umfasst zu veranlassen, dass die erste Speicherungseinrichtung (**105**) bestätigt, dass Daten, die mit der bezeichneten zweiten Schreibanforderung assoziiert sind, in die nichtflüchtige Speicherung der ersten Speicherungseinrichtung (**105**) geschrieben worden sind.

15. Verfahren nach einem der Ansprüche 12 bis 14, weiterhin umfassend zu veranlassen, dass die erste Speicherungseinrichtung (**105**) nach dem Empfangen der zweiten Commit-Synchronisierungs-Nachricht (**160**) die bezeichnete zweite Schreibanforderung verarbeitet.

16. Verfahren nach einem der Ansprüche 12 bis 15, wobei zu veranlassen, dass die erste Speicherungseinrichtung (**105**) bestätigt, dass die mit der bezeichneten zweiten Schreibanforderung assoziierten Daten in die nichtflüchtige Speicherung der ersten Speicherungseinrichtung (**105**) geschrieben worden sind, umfasst zu veranlassen, dass die erste Speicherungseinrichtung (**105**) eine erfolgreiche Cache-Leerung der flüchtigen Speicherung der zweiten Speicherungseinrichtung (**110**) bestätigt.

17. Verfahren nach einem der Ansprüche 12 bis 16, wobei es sich bei den mit der zweiten Commit-Synchronisierungs-Nachricht (**160**) gesendeten Informationen um eine Referenzmarke handelt, die eine von der zweiten Speicherungseinrichtung (**110**) verarbeitete zweite Schreibanforderung identifiziert.

18. Verfahren nach Anspruch 17, wobei die Referenzmarken sequenziell Schreibanforderungen (**130**) zugeordnet werden.

19. Verfahren nach Anspruch 18, wobei das Verarbeiten der bei der ersten Speicherungseinrichtung (**105**) empfangenen Schreibanforderungen (**130**) weiterhin das Verarbeiten von Schreibanforderungen (**130**) in sequenzieller Reihenfolge nach ihren Referenzmarken derart umfasst, dass alle Schreibanforderungen (**130**), die vor der von der Referenzmarke in der zweiten Commit-Synchronisierungs-Nachricht (**160**) identifizierten Schreibanforderung ausgegeben werden, vor der Verarbeitung dieser Schreibanforderung verarbeitet werden.

20. Verfahren nach Anspruch 18 oder 19, wobei jede Speicherungseinrichtung Schreibanforderungen

(130) mit der gleichen Sequenz von Referenzmarken empfängt.

21. Verfahren nach einem der Ansprüche 12 bis 20, weiterhin umfassend das Identifizieren von Speicherungsgebieten, die von Schreibanforderungen (130) betroffen sind, die bei der zweiten Speicherungseinrichtung (110) verarbeitet worden sind.

22. Verfahren nach Anspruch 21, wobei das Identifizieren von Speicherungsgebieten, die von Schreibanforderungen (130) betroffen sind, die bei der zweiten Speicherungseinrichtung (110) verarbeitet worden sind, weiterhin das Akkumulieren der identifizierten Speicherungsgebiete in einer vierten Bitmap umfasst.

23. Verfahren nach Anspruch 22, weiterhin umfassend: Akkumulieren von neu identifizierten Speicherungsgebieten in einer fünften Bitmap nach dem Senden der zweiten Commit-Synchronisierungs-Nachricht (160), Senden einer Statusnachricht an die zweite Speicherungseinrichtung (110), die anzeigt, ob die Schreibdaten erfolgreich in die nichtflüchtige Speicherung geschrieben wurden, nachdem die erste Speicherungseinrichtung (105) bestätigt, dass Daten in den verarbeiteten Schreibanforderungen (130) in eine nichtflüchtige Speicherung der ersten Speicherungseinrichtung (105) geschrieben worden sind, und Löschen der vierten Bitmap nach dem Empfangen der zweiten Statusnachricht, die anzeigt, dass die Schreibdaten erfolgreich geschrieben wurden.

24. Verfahren nach Anspruch 23, weiterhin umfassend das Kopieren des Inhalts der fünften Bitmap in die vierte Bitmap und Löschen der fünften Bitmap nach dem Empfangen der Statusnachricht, die anzeigt, dass die Schreibdaten nicht erfolgreich geschrieben wurden.

25. Verfahren nach Anspruch 23, weiterhin umfassend das Bezeichnen der fünften Bitmap als der vierten Bitmap nach dem Löschen der vierten Bitmap.

26. Verfahren nach Anspruch 22, weiterhin umfassend, nach einer Periode, wenn die erste Speicherungseinrichtung (105) nicht in der Lage war, Schreibanforderungen (130) zu verarbeiten: Kopieren des Inhalts der vierten Bitmap in eine zweite Wiederherstellungsbitmap, Verwenden der zweiten Wiederherstellungsbitmap zum Identifizieren der Speicherungsgebiete der zweiten Speicherungseinrichtung (110), die von der zweiten Speicherungseinrichtung (110) zu der ersten Speicherungseinrichtung (105) kopiert werden sollen, Kopieren der identifizierten Speicherungsgebiete der zweiten Speicherungseinrichtung (110) zu der ersten

Speicherungseinrichtung (105) und Akkumulieren neu empfangener Schreibanforderungen (130) bei der ersten Speicherungseinrichtung (105) in einer sechsten Bitmap.

27. Verfahren nach einem der vorhergehenden Ansprüche, weiterhin umfassend das Assoziieren einer eindeutigen Kennung mit einem bestimmten Datensatz, so dass die Schreibanforderung eindeutig identifiziert ist.

28. Verfahren nach Anspruch 27, wobei die eindeutige Kennung eine Fallnummer oder ein Datum umfasst.

29. Verfahren nach Anspruch 1, bei dem die erste Speicherungseinrichtung (105) weiterhin eine flüchtige Speicherung und eine nichtflüchtige Speicherung enthält, wobei das Verfahren umfasst:

Senden der Commit-Synchronisierungs-Nachricht zusammen mit einer Schreibanforderung bezeichnenden Informationen an die zweite Speicherungseinrichtung (110) nach dem Bestimmen, dass die zweite Speicherungseinrichtung (110) dabei ist, in eine Periode einzutreten, in der die zweite Speicherungseinrichtung (110) nicht in der Lage sein wird, Schreibanforderungen (130) zu verarbeiten;

nach dem Senden der Commit-Synchronisierungs-Nachricht (160) zu veranlassen, dass der Controller (115) der ersten Speicherungseinrichtung (105) Speicherungsgebiete, die von neuen Schreibanforderungen (130) betroffen sind, in einer Bitmap akkumuliert;

nachdem die zweite Speicherungseinrichtung (110) in der Lage ist, Schreibanforderungen (130) zu verarbeiten, zu veranlassen, dass der Controller (115) der ersten Speicherungseinrichtung (105) die Bitmap verwendet, um die Speicherungsgebiete der ersten Speicherungseinrichtung (105) zu identifizieren, die von der ersten Speicherungseinrichtung (105) zu der zweiten Speicherungseinrichtung (110) kopiert werden sollen; und

Kopieren des Inhalts der identifizierten Gebiete der ersten Speicherungseinrichtung (105) zu der zweiten Speicherungseinrichtung (110).

30. Verfahren nach Anspruch 1, bei dem die erste Speicherungseinrichtung (105) weiterhin eine flüchtige Speicherung und eine nichtflüchtige Speicherung enthält, wobei das Verfahren umfasst:

Veranlassen, dass der Controller (115) der ersten Speicherungseinrichtung (105) von neuen Schreibanforderungen (130) betroffene Speicherungsgebiete in einer ersten Bitmap (155) akkumuliert,

Veranlassen, dass der Controller (115) der ersten Speicherungseinrichtung (105) die Commit-Synchronisierungs-Nachricht (160) zusammen mit Informationen, die eine eindeutig identifizierte Schreibanforderung bezeichnen, an die zweite Speicherungsein-

richtung (110) sendet;
 Veranlassen, dass der Controller (120) der zweiten Speicherungseinrichtung (110) nach dem Empfangen der Commit-Synchronisierungs-Nachricht (160) bestätigt, dass mit der bezeichneten Schreibanforderung assoziierte Daten in die nichtflüchtige Speicherung der zweiten Speicherungseinrichtung (110) geschrieben worden sind;
 nach dem Senden der Commit-Synchronisierungs-Nachricht (160) zu veranlassen, dass der Controller (115) der ersten Speicherungseinrichtung (105) von neuen Schreibanforderungen (130) betroffene Speicherungsgebiete in einer zweiten Bitmap (156) akkumuliert;
 Senden einer Statusnachricht an die erste Speicherungseinrichtung (105), die anzeigt, ob die Schreibdaten erfolgreich in eine nichtflüchtige Speicherung geschrieben wurden, nachdem der Controller (120) der zweiten Speicherungseinrichtung (110) bestätigt, dass Daten in den verarbeiteten Schreibanforderungen (130) zu einer nichtflüchtigen Speicherung der zweiten Speicherungseinrichtung (110) geschrieben worden sind;
 Löschen der ersten Bitmap (155) nach dem Empfangen der Statusnachricht, die anzeigt, dass die Schreibdaten erfolgreich geschrieben wurden;
 Kopieren des Inhalts der zweiten Bitmap (156) in die erste Bitmap (155) und Löschen der zweiten Bitmap (156) nach dem Empfangen der Statusnachricht, die anzeigt, dass die Schreibdaten nicht erfolgreich geschrieben wurden;
 nachdem die zweite Speicherungseinrichtung (110) in der Lage ist, Schreibanforderungen (130) nach einer Periode zu verarbeiten, in der die zweite Speicherungseinrichtung (110) nicht in der Lage war, Schreibanforderungen (130) zu verarbeiten:
 Kopieren des Inhalts der zweiten Bitmap (156) in eine Wiederherstellungsbitmap,
 Verwenden der Wiederherstellungsbitmap zum Identifizieren von Speicherungsgebieten der ersten Speicherungseinrichtung (105), die von der ersten Speicherungseinrichtung (105) zu der zweiten Speicherungseinrichtung (110) kopiert werden sollen,
 Kopieren des Inhalts der identifizierten Speicherungsgebiete der ersten Speicherungseinrichtung (105) zu der zweiten Speicherungseinrichtung (110) und
 Veranlassen, dass der Controller (115) der ersten Speicherungseinrichtung (105) von neuen Schreibanforderungen (130) betroffene Speicherungsgebiete in einer dritten Bitmap (165) akkumuliert.

31. Verfahren nach Anspruch 30, wobei das Kopieren des Inhalts der identifizierten Gebiete der ersten Speicherungseinrichtung (105) zu der zweiten Speicherungseinrichtung (110) Folgendes umfasst:
 Veranlassen, dass der Controller (115) der ersten Speicherungseinrichtung (105) eine Commit-Synchronisierungs-Nachricht (160) zusammen mit Infor-

mationen, die eine eindeutig identifizierte Schreibanforderung bezeichnen, an die zweite Speicherungseinrichtung (110) sendet;
 Veranlassen, dass der Controller (120) der zweiten Speicherungseinrichtung (110) nach dem Empfangen der Commit-Synchronisierungs-Nachricht (160) bestätigt, dass mit der bezeichneten Schreibanforderung assoziierte Daten in die nichtflüchtige Speicherung der zweiten Speicherungseinrichtung (110) geschrieben worden sind;
 nach dem Senden der Commit-Synchronisierungs-Nachricht (160) zu veranlassen, dass der Controller (115) der ersten Speicherungseinrichtung (105) Speicherungsgebiete, die von neuen Schreibanforderungen (130) betroffen sind, in einer vierten Bitmap akkumuliert;
 nachdem die zweite Speicherungseinrichtung (110) bestätigt, dass Daten in den verarbeiteten Schreibanforderungen (130) in eine nichtflüchtige Speicherung der zweiten Speicherungseinrichtung (110) geschrieben worden sind, Senden einer Statusnachricht an die erste Speicherungseinrichtung (105), die anzeigt, ob die Schreibdaten erfolgreich in die nichtflüchtige Speicherung geschrieben wurden;
 Löschen der dritten Bitmap (165) nach dem Empfangen der Statusnachricht, die anzeigt, dass die Schreibdaten erfolgreich geschrieben wurden; und
 Kopieren des Inhalts der vierten Bitmap zur dritten Bitmap (165) und Löschen der vierten Bitmap nach dem Empfangen der Statusnachricht, die anzeigt, dass die Schreibdaten nicht erfolgreich geschrieben wurden.

32. Gespiegeltes Datenspeicherungssystem, umfassend:
 eine erste Speicherungseinrichtung (105);
 eine zweite Speicherungseinrichtung (110);
 einen mit der ersten Speicherungseinrichtung (105) assoziierten ersten Controller (115) und
 einen mit der zweiten Speicherungseinrichtung (110) assoziierten zweiten Controller (120);
 wobei:
 der erste Controller (115) konfiguriert ist:
 Schreibanforderungen (130) bei einer ersten Speicherungseinrichtung (105) zu empfangen;
 die bei der ersten Speicherung empfangenen Schreibanforderungen (130) zu verarbeiten und
 eine Commit-Synchronisierungs-Nachricht (160) zusammen mit Informationen, die eine Schreibanforderung bezeichnen, an die zweite Speicherungseinrichtung (110) zu senden, wobei es sich bei den mit der Commit-Synchronisierungs-Nachricht (160) gesendeten Informationen um eine Referenzmarke handelt, die eine von der ersten Speicherungseinrichtung (105) verarbeitete Schreibanforderung identifiziert, wobei die Referenzmarken sequenziell Schreibanforderungen (130) zugeordnet sind; und der zweite Controller (120) konfiguriert ist:
 Schreibanforderungen (130) bei einer zweiten Speicherungseinrichtung (110) zu empfangen, wobei die

zweite Speicherungseinrichtung (110) flüchtige Speicherung und nichtflüchtige Speicherung enthält; die bei der zweiten Speicherungseinrichtung (110) empfangenen Schreibanforderungen (130) zu verarbeiten und

nach dem Empfangen der Commit-Synchronisierungs-Nachricht und Verarbeiten der durch die Informationen in der Commit-Synchronisierungs-Nachricht identifizierten Schreibanforderung zu bestätigen, dass Daten, die mit allen Schreibanforderungen (130) assoziiert sind, die der bezeichneten Schreibanforderung vorausgehen, in die nichtflüchtige Speicherung der zweiten Speicherungseinrichtung (110) geschrieben worden sind und dass Daten, die mit der durch die Informationen in der Commit-Synchronisierungs-Nachricht (160) identifizierten Schreibanforderung assoziiert sind, in die nichtflüchtige Speicherung der zweiten Speicherungseinrichtung (110) geschrieben worden sind.

33. System nach Anspruch 32, wobei der zweite Controller (120) konfiguriert ist, eine erfolgreiche Cache-Leerung der flüchtigen Speicherung der zweiten Speicherungseinrichtung (110) zu bestätigen.

34. System nach Anspruch 32 oder 33, wobei der zweite Controller (120) konfiguriert ist, Schreibanforderungen (130) in sequenzieller Reihenfolge nach ihren Referenzmarken zu verarbeiten, so dass alle Schreibanforderungen (130), die vor der durch die Referenzmarke in der Commit-Synchronisierungs-Nachricht (160) identifizierten Schreibanforderung ausgegeben werden, vor der Verarbeitung dieser Schreibanforderung verarbeitet werden.

35. System nach einem der Ansprüche 32 bis 34, wobei jede Speicherungseinrichtung Schreibanforderungen (130) mit der gleichen Sequenz von Referenzmarken empfängt.

36. System nach einem der Ansprüche 32 bis 35, wobei der erste Controller (115) weiterhin konfiguriert ist, Speicherungsgebiete zu identifizieren, die von Schreibanforderungen (130) betroffen sind, die bei der ersten Speicherungseinrichtung (105) verarbeitet worden sind.

37. System nach Anspruch 36, wobei der erste Controller (115) konfiguriert ist, die identifizierten Speicherungsgebiete in einer ersten Bitmap (155) zu akkumulieren.

38. System nach Anspruch 36 oder 37, wobei der zweite Controller (120) weiterhin konfiguriert ist, nachdem die zweite Speicherungseinrichtung (110) bestätigt, dass Daten in den verarbeiteten Schreibanforderungen (130) in eine nichtflüchtige Speicherung der zweiten Speicherungseinrichtung (110) geschrieben worden sind, eine Statusnachricht an die erste Speicherungseinrichtung (105) zu senden, die an-

zeigt, ob die Schreibdaten erfolgreich in die nichtflüchtige Speicherung geschrieben wurden, wobei der erste Controller (115) weiterhin konfiguriert ist, nach dem Senden der Commit-Synchronisierungs-Nachricht (160) neu identifizierte Speicherungsgebiete in einer zweiten Bitmap (156) zu akkumulieren und nach dem Empfangen der Statusnachricht, die anzeigt, dass die Schreibdaten erfolgreich geschrieben wurden, die erste Bitmap (155) zu löschen.

39. System nach Anspruch 38, wobei der erste Controller (115) weiterhin konfiguriert ist, nach dem Empfangen der Statusnachricht, die anzeigt, dass die Schreibdaten nicht erfolgreich geschrieben wurden, den Inhalt der zweiten Bitmap (156) in die erste Bitmap (155) zu kopieren und die zweite Bitmap (156) zu löschen.

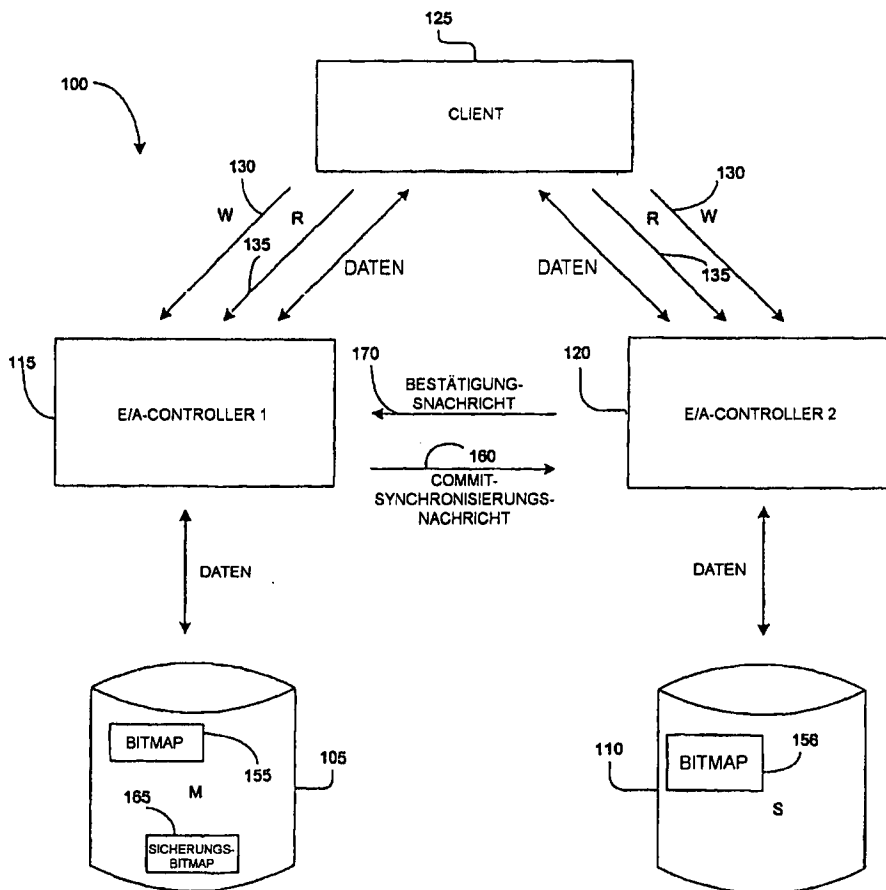
40. System nach Anspruch 37, wobei der erste Controller (115) weiterhin konfiguriert ist, nach einer Periode, wenn die zweite Speicherungseinrichtung (110) nicht in der Lage war, Schreibanforderungen (130) zu verarbeiten:

den Inhalt der ersten Bitmap (155) in eine Wiederherstellungsbitmap zu kopieren, die Wiederherstellungsbitmap zum Identifizieren von Speicherungsgebieten der ersten Speicherungseinrichtung (105) zu verwenden, die von der ersten Speicherungseinrichtung (105) zu der zweiten Speicherungseinrichtung (110) kopiert werden sollen, die identifizierten Speicherungsgebiete der ersten Speicherungseinrichtung (105) zu der zweiten Speicherungseinrichtung (110) zu kopieren und neu empfangene Schreibanforderungen (130) bei der ersten Speicherungseinrichtung (105) in einer dritten Bitmap (165) zu akkumulieren.

41. Computerlesbares Medium oder ausgebreitetes Signal mit darauf verkörpert einem Computerprogramm, das konfiguriert ist, eine gespiegelte Kopie einer ersten Speicherungseinrichtung (105) bei einer zweiten Speicherungseinrichtung (110) in einem Computersystem zu unterhalten, wobei das Medium Codesegmente umfasst, die konfiguriert sind, das Verfahren nach einem der Ansprüche 1 bis 31 durchzuführen, wenn es auf einem Computer ausgeführt wird.

Es folgen 7 Blatt Zeichnungen

Anhängende Zeichnungen



200

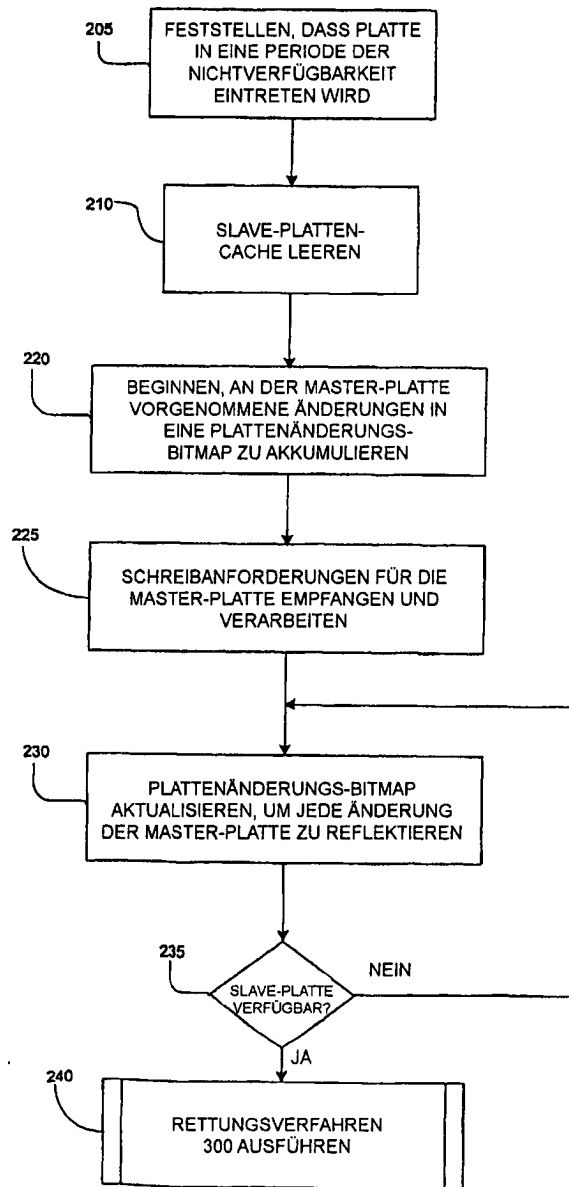


FIG. 2

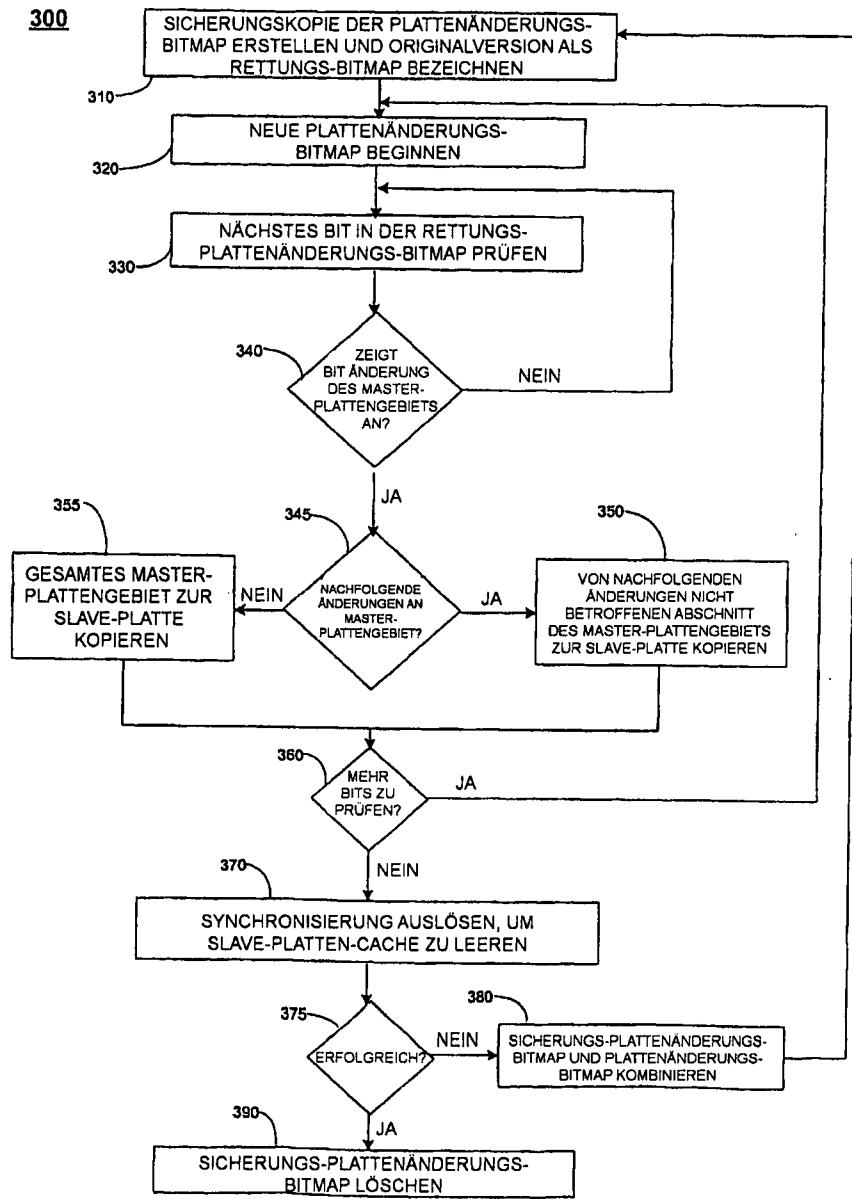


FIG. 3

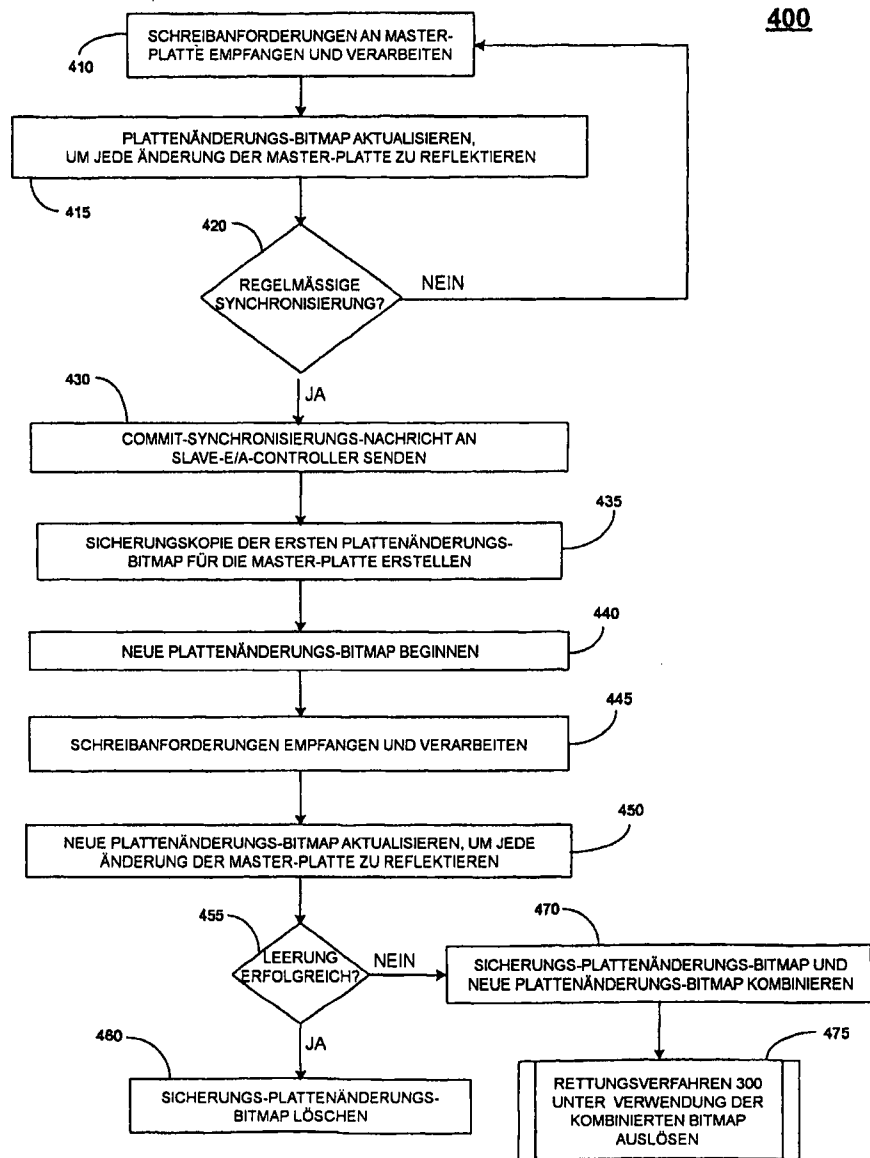


FIG. 4

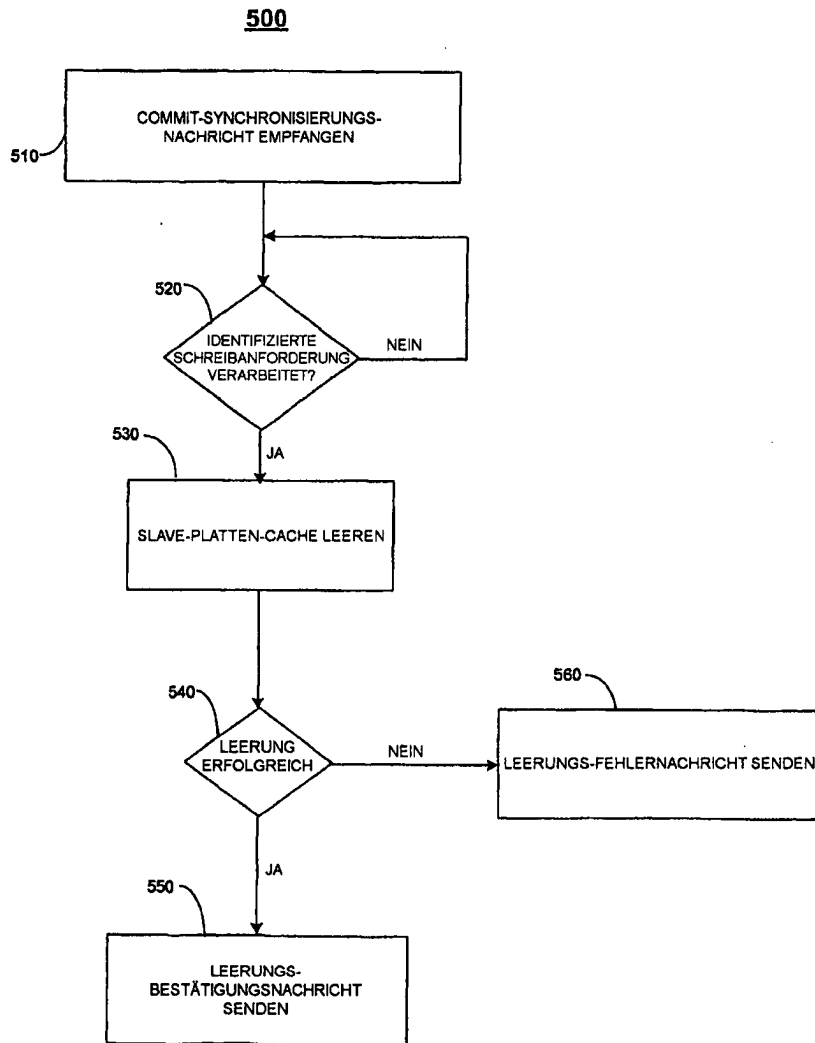


FIG. 5

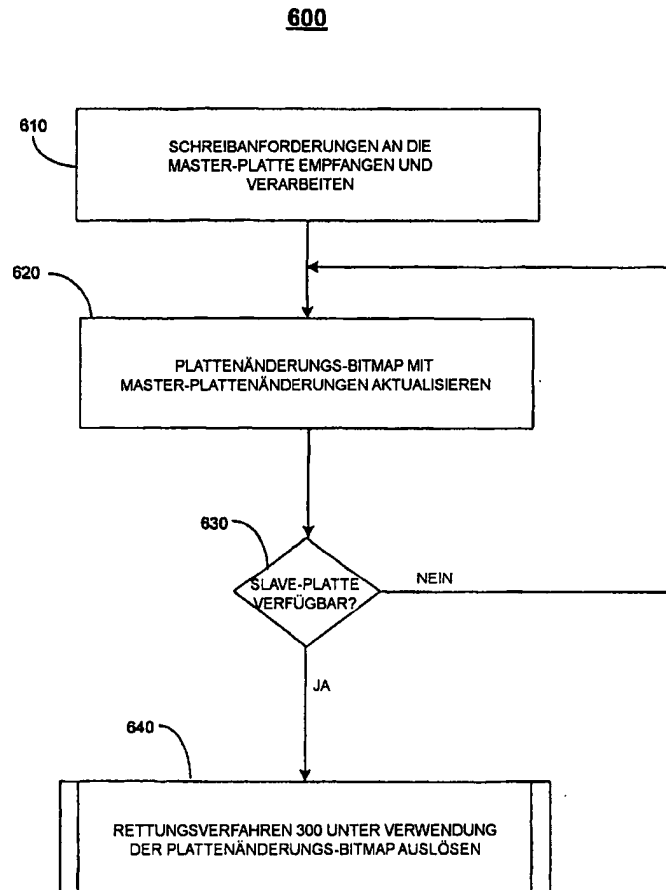


FIG. 6

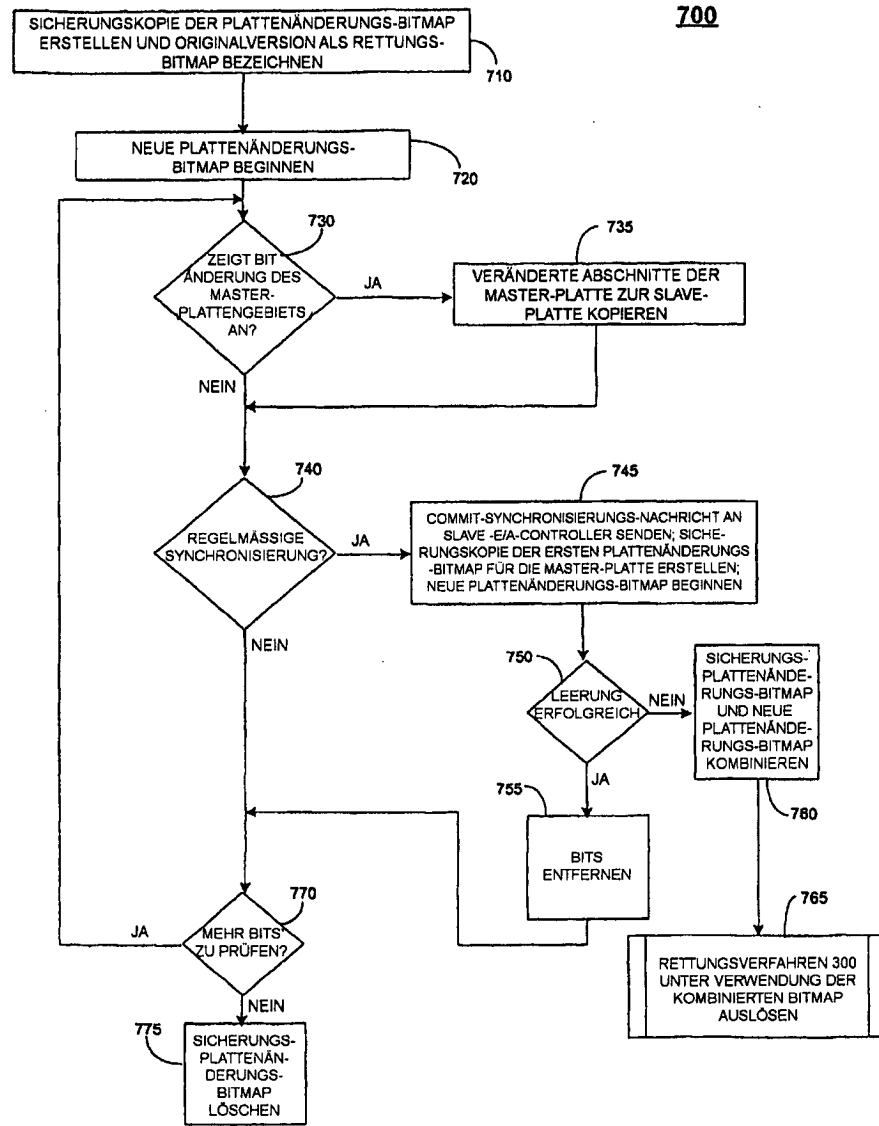


FIG. 7