



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 102 09 803 B4** 2004.07.29

(12)

Patentschrift

(21) Aktenzeichen: **102 09 803.4**
(22) Anmeldetag: **06.03.2002**
(43) Offenlegungstag: **17.10.2002**
(45) Veröffentlichungstag
der Patenterteilung: **29.07.2004**

(51) Int Cl.⁷: **G06F 3/06**

Innerhalb von 3 Monaten nach Veröffentlichung der Erteilung kann Einspruch erhoben werden.

(30) Unionspriorität:
819265 **27.03.2001** **US**

(71) Patentinhaber:
**Hewlett-Packard Co. (n.d.Ges.d.Staates
Delaware), Palo Alto, Calif., US**

(74) Vertreter:
**Schoppe, Zimmermann, Stöckeler & Zinkler, 82049
Pullach**

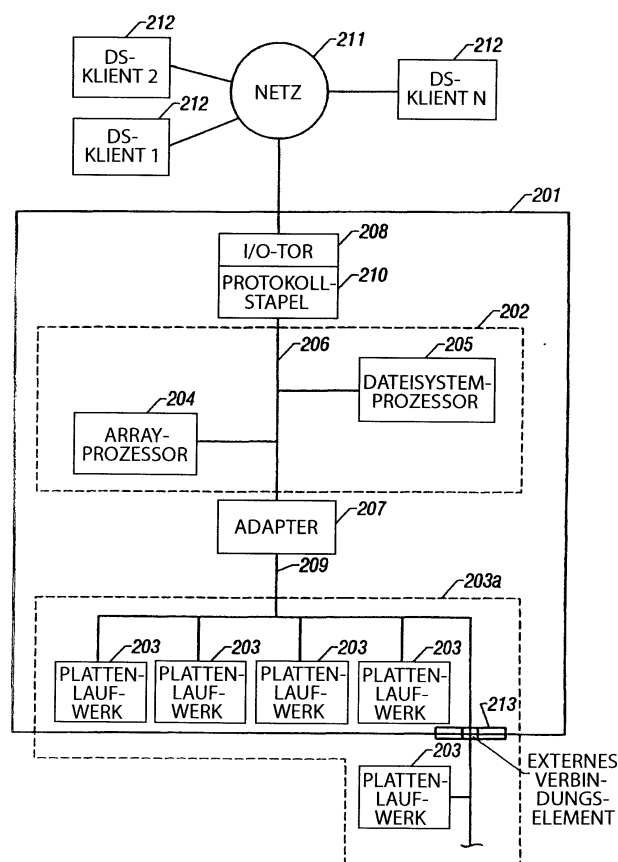
(72) Erfinder:
Voigt, Douglas L., Boise, Id., US

(56) Für die Beurteilung der Patentfähigkeit in Betracht
gezogene Druckschriften:
US 58 75 459 A
EP 12 21 646 A2

(54) Bezeichnung: **Verfahren, Datenspeichersystem und Computerprogramm zum Liefern eines Dateisystemzugriffs auf ein Plattenarray**

(57) Hauptanspruch: Verfahren zum Zugreifen auf ein Datenspeichergerät (203), das durch Arraysteuerungsprozesse (300) gesteuert wird, wobei das Verfahren folgende Schritte umfaßt:

- (a) Empfangen einer Dateisystemzugriffsoperationsanforderung;
- (b) Bestimmen, ob die Dateisystemzugriffsoperationsanforderung eine Operation spezifiziert, die eine umleitbare Operation aufweist;
- (c) falls die Dateisystemzugriffsoperationsanforderung eine umleitbare Operation spezifiziert, Umleiten der umleitbaren Operation von Dateisystemprozessen (308), die dem Datenspeichergerät (203) zugeordnet sind, zu den Arraysteuerungsprozessen (300);
- (d) Durchführen der umleitbaren Operation mit den Arraysteuerungsprozessen (300); und
- (e) Aktualisieren einer Dateisystemdatenverwaltungsanordnung mit Operationsergebnisinformationen von den Arraysteuerungsprozessen, wobei die Dateisystemdatenverwaltungsanordnung durch die Dateisystemprozesse (308) gesteuert wird.



Beschreibung

[0001] Diese Erfindung bezieht sich auf Datenspeichersysteme, die mit Computern und Computernetzen verwendet werden. Insbesondere bezieht sich die Erfindung auf ein Verfahren und ein Computerprogramm zum Beschleunigen eines Dateisystemzugriffs auf ein Plattenarray. Die Erfindung umfaßt sowohl ein Datenspeichersystem als auch ein Verfahren zum Zugreifen auf eine Datenspeichervorrichtung. Außerdem umfaßt die Erfindung ein Programmprodukt zum Bedienen von Dateisystemzugriffsoperationen von einem Dateisystemklienten.

Stand der Technik

[0002] Dateisysteme oder „File-Systeme“ und Plattenarrays verwenden zwei unterschiedliche Verfahren zum Zugreifen auf Daten, die auf einem Datenspeichermedium gespeichert sind. Die beiden Verfahren unterscheiden sich durch eine Zahl von Charakteristika. Bei einem Dateisystem wird z. B. auf Dateien unter Verwendung einer Dateiabstraktion zugegriffen. Dies bedeutet, daß Daten durch eine Datei dargestellt werden, die geöffnet, geschlossen, erzeugt, gelöscht, gelesen und geschrieben werden kann. Daten werden in einem Dateisystem durch eine Hierarchie von Dateien und Verzeichnissen verwaltet. Diese hierarchische Datenverwaltung wird allgemein als ein wesentlicher Vorteil gegenüber anderen Datenverwaltungsschemata betrachtet. Dateisysteme unterstützen auch sowohl eine gemeinschaftliche Verwendung von Daten als auch eine gemeinschaftliche Verwendung eines Datenspeichergeräts. Die Sicherheit in Datenspeichergeräten basiert auf Datei- und Verzeichnispegelzugriffssteuerungslisten, die als ein Teil der Dateisystemprotokolle standardisiert sind. Ein weiterer Vorteil von Dateisystemen besteht darin, daß eine Klient/Server-Kompatibilität auf einer kleinen Zahl von gut unterstützten Protokollen basiert. Die Dateisystemleistung wird jedoch durch Dateiabbildungsanordnungen, die sich auf einer Datenspeicherplatte befinden, reduziert, wobei dieselben Daten für eine Datei an physischen oder virtuellen Orten des Speichermediums, das einem Datenspeichergerät zugeordnet ist, abbilden müssen.

[0003] Ein Plattenarray ist ein Speichersystem, das eine Zahl von separaten Speichergeräten oder -Platten verwendet und Daten über alle der separaten Geräte verwaltet. Diese Plattenarrayspeichersysteme sind zum Implementieren von Datenspeicherredundanzschemata weit verbreitet, um vor Fehlern einzelner Speichergeräte zu schützen. Im Gegensatz zu einem Dateisystem wird auf Daten in einem Plattenarray unter Verwendung einer Blockspeicherabstraktion zugegriffen. Dies bedeutet, daß Daten auf große zusammenhängende Speicherräume oder logische Einheiten abgebildet werden, und daß auf dieselben als Blöcke, die in diesen logischen Einheiten enthal-

ten sind, zugegriffen wird. Jeder Block in einer logischen Einheit stellt eine bestimmte Menge von Datenspeicherraum dar, üblicherweise einen Speicherraum von 512 Bytes. Die gespeicherten Daten in einem Plattenarray werden als physische und virtuelle Datenspeichergeräte verwaltet, die durch Volumina und/oder logischer Einheiten dargestellt sind. Obwohl die Blockspeicherabstraktion die Geschwindigkeit verbessern kann, mit der große Datenmengen geschrieben und von den Speichergeräten wiedergewonnen werden können, gibt es eine Zahl von Nachteilen, die Plattenarrays verglichen mit Dateisystemen zugeordnet sind. Ein Nachteil besteht darin, daß die Klient/Server-Kompatibilität für Plattenarrays einer Vielzahl von Beschränkungen unterworfen ist, die unter den Klienten und Servern variieren. Außerdem unterstützen Plattenarrays nur eine gemeinschaftliche Verwendung des Geräts, nicht jedoch die der Daten. Ferner basiert die Sicherheit in Geräten auf lieferantenspezifischen Merkmalen.

[0004] **Fig. 1** liefert eine schematische Darstellung einer Anordnung des Stands der Technik zwischen einem Plattenarraysystem **100** und einem Dateisystem **101**. Die Anordnung umfaßt ein Array physischer Speichergeräte **102**, die Plattenlaufwerke sein können, die jeweils zur Kommunikation mit einer Arraysteuerung **103** verbunden sind. Die Arraysteuerung **103** ist ein Verarbeitungsgerät, das Arraysteuerungsprogrammprozesse ausführt, um Geräte **102** zu steuern und die Daten, die über die Geräte gespeichert sind, zu verwalten. Die Arraysteuerung **103** kommuniziert mit einem Hostcomputer **104** durch eine geeignete Kommunikationsanordnung, die durch eine Linie **105** dargestellt ist. Das Dateisystem **101** ist durch Computerprogrammprozesse implementiert, die durch einen Prozessor an dem Hostcomputer **104** ausgeführt werden. Obwohl **Fig. 1** die Arraysteuerung separat von dem Hostcomputer zeigt, kombinieren einige Plattenarraysysteme des Stands der Technik die Funktionen der Arraysteuerung und des Hostprozessors in einem einzelnen Verarbeitungsgerät, das dem Plattenarraysystem zugeordnet ist. Die Kooperation zwischen der Plattenarraysteuerung **103** und dem Dateisystem **101** ist im allgemeinen die gleiche, und zwar unabhängig davon, ob die Arraysteuerungs- und die Dateisystemprozesse auf einem einzelnen Verarbeitungsgerät oder separaten Verarbeitungsgeräten durchgeführt werden.

[0005] Bei dem System **100** des Stands der Technik, das in **Fig. 1** gezeigt ist, kann man sich das Dateisystem und Plattenarraysteuerungsprozesse als zwei separate Verarbeitungsschichten vorstellen, die kooperieren, um einen Zugriff auf das physische Medium, das in Speichergeräten **102** enthalten ist, zu liefern. Das Dateisystem **101** ist im wesentlichen zwischen den Dateisystemklienten (in **Fig. 1** nicht gezeigt) und der Plattenarraysteuerung **103** eingefügt, die tatsächlich einen Zugriff auf die Geräte **102** verwaltet und steuert. Um zum Beispiel eine Datei zu lesen, legt ein Dateisystemklient dem Dateisystem **101**

eine Lesen-Operation-Anforderung vor. Dieser Dateisystemklient weist einen Programmprozeß auf, der an dem Hostcomputer **104** oder an einem Computer, der netzmäßig mit dem Hostcomputer verbunden ist, ausgeführt werden kann. In jedem Fall empfangen Dateisystemprozesse die Anforderung von dem Dateisystemklienten und aktualisieren die Dateisystemdatenverwaltungsanordnung, die den Dateinamen, üblicherweise verschiedene Dateiattribute, und Speicherzuweisungsinformationen speichert, die anzeigen, wo die Daten, die die Datei ausmachen, gespeichert sind. Das Dateisystem **101** gibt dann eine geeignete Instruktion an die Plattenarraysteuerung **103** aus. Die Arraysteuerung **103** verarbeitet die Instruktion, um die angeforderten Dateidaten von einem oder mehreren Speichergeräten **102** zu lesen. Die Daten, die von Speichergeräten **102** gelesen werden, werden zuerst in einen Speicher Cache-gespeichert, der der Arraysteuerung **103** zugeordnet ist, und dann zu einem Cachespeicher geleitet, der dem Dateisystem **101** zugeordnet ist. Das Dateisystem **101** gibt dann die angeforderten Daten wie durch die Lesen-Operation angeleitet zurück. Dieser Schritt des Übertragens von Daten von der Plattenarraysteuerung **103** zu dem Dateisystem **101**, bevor die Daten zu dem Dateisystem übertragen werden können, bringt einen Leistungsnachteil mit sich, insbesondere wenn die Datei, auf die zugegriffen wird, eine große Datei ist.

[0006] In der EP 1221646A, welche ein nachveröffentlichtes Dokument darstellt, wird vorgeschlagen, eine Diskettenarraysteuerung zu aktivieren, um kontinuierlich Daten, für die ein Schreiben angefordert wurde, zu einem Streifen auf einer Diskette unter Verwendung von nur dem Hauptspeicher und einer Diskettenvorrichtung zu schreiben, ohne daß ein nicht-flüchtiger Speicher zum Sammeln der zu dem Streifen zu schreibenden Daten verwendet würde. Der in der EP 1221646 beschriebene Prozeß verwendet jedoch immer noch das grundsätzliche Modell der Zusammenarbeit zwischen den Dateisystemprozessen und einem Diskettenarraysteuerungselement, das oben anhand der **Fig. 1** beschrieben wurde. Das heißt, daß gemäß der EP 1221646 die Schreiboperationen immer noch durch das Dateisystem verarbeitet werden, welches seinerseits die Informationen an die Diskettenarraysteuerung weiterleitet, die für die Diskettenarraysteuerung erforderlich sind, um die tatsächliche Schreiboperation auf dem Medium durchzuführen.

[0007] In der US-A-5,875,459 wurde vorgeschlagen, alternativ bestimmte Dateien auf einer optischen Scheibenarrayvorrichtung auf einer einzelnen optischen Scheibe oder auf einer Anzahl von optischen Scheiben zu speichern, abhängig von bestimmten Charakteristika der Datei. Der Zweck der Speicherung bestimmter Dateien über mehrere optische Scheiben in dieser Schrift besteht darin, daß die mehrfachen optischen Scheiben parallel zum Lesen oder Zugreifen auf die Dateidaten betrieben werden können, was die Datenzugriffszeit reduziert. Die

US-A-5,875,459 beschreibt jedoch keine Anordnung zum Umleiten von Dateizugriffsanfragen, um die normalen Dateisystemprozesse herum, sondern lediglich das Speichern einer Datei auf mehreren optischen Laufwerken, wenn ein schneller Zugriff auf die Daten erwünscht ist.

Aufgabenstellung

[0008] Es ist die Aufgabe der vorliegenden Erfindung, ein verbessertes Verfahren zum Zugreifen auf ein Datenspeichergerät, ein Computerprogramm mit verbesserten Charakteristika oder ein Datenspeichersystem mit verbesserten Charakteristika, zu schaffen.

[0009] Diese Aufgabe wird durch ein Verfahren zum Zugreifen auf ein Datenspeichergerät gemäß Anspruch 1, ein Computerprogramm gemäß Anspruch 15 oder 21 oder ein Datenspeichersystem gemäß Anspruch 26 gelöst.

[0010] Folglich besteht ein Vorteil dieser Erfindung darin, ein Verfahren und eine Vorrichtung zu schaffen, die die oben beschriebenen Probleme, die Datenspeichersystemen zugeordnet sind, überwinden. Insbesondere besteht ein Vorteil dieser Erfindung darin, ein Verfahren und eine Vorrichtung zu schaffen, die bestimmte Dateisystemzugriffsoperationen außerhalb des Dateisystems durchführen, während die hierarchische Struktur des Dateisystems aufrechterhalten wird. Außerdem besteht ein Vorteil der vorliegenden Erfindung darin, ein Programmprodukt zum Bedienen von Dateisystemzugriffsoperationsanforderungen auf eine Weise zu schaffen, die den Dateisystemzugriff auf Dateien, die durch eine Plattenarrayimplementierung gespeichert sind, beschleunigt.

[0011] Diese Vorteile werden bei einem System, das Dateisystemprozesse, Arraysteuerungsprozesse und eines oder mehrere Datenspeichergeräte umfaßt, auf die durch die Arraysteuerungsprozesse zugegriffen wird, erzielt. Gemäß der Erfindung werden bestimmte Dateisystemzugriffsoperationen von den Dateisystemprozessen umgeleitet, so daß die Arraysteuerungsprozesse direkt die angeforderte Zugriffsoperation durchführen können. Die Arraysteuerungsprozesse kommunizieren dann mit den Dateisystemprozessen, so daß die Dateisystemprozesse die Dateisystemdatenverwaltungsanordnung geeignet aktualisieren können. Auf diese Weise wird die erwünschte hierarchische Dateisystemverwaltung beibehalten, während auf das Speichermedium direkt durch die Arraysteuerungsprozesse ansprechend auf eine Dateisystemzugriffsoperationsanfrage zugegriffen werden kann. Dieser direkte Speichergeräatzugriff für Dateisystemzugriffsoperationen ist insbesondere für große Dateien von Vorteil.

[0012] Der Ausdruck „Dateisystemprozesse“, wie er in dieser Offenbarung verwendet wird, bezieht sich auf Software oder einen Computerprogrammcode, der durch ein geeignetes Verarbeitungsgerät ausgeführt wird, um das Dateisystem zu implementieren.

Dieser Programmcode umfaßt einen Code zum Implementieren einer Datenverwaltungsanordnung für das Dateisystem. Die Dateisystemdatenverwaltungsanordnung umfaßt Informationen für jede Datei, die durch das Dateisystem verwaltet wird, wobei diese Informationen in einer oder mehreren Datentabellen gespeichert sein können, die den Dateinamen auf verschiedene Attribute für die Datei und auf Speicherzuteilungsinformationen für die Datei beziehen.

[0013] Der Ausdruck „Arraysteuerungsprozesse“ bezieht sich auf die Software oder einen Computerprogrammcode, der durch ein geeignetes Verarbeitungsgerät zum Steuern und Verwalten eines Arrays von Platten oder anderen physischen Speichergeräten ausgeführt wird. Diese Arraysteuerungsprozesse umfassen einen Programmcode zum Interpretieren eines empfangenen Befehls und einen Abbildungscode zum Definieren logischer Einheiten in den physischen Speichergeräten und zum Abbilden von Daten an den verschiedenen logischen Einheiten und an Blöcken, die in den verschiedenen logischen Einheiten enthalten sind. Ein Beispiel von Arraysteuerungsprozessen ist in dem U.S.-Patent Nr. 5,392,244 beschrieben, wobei der gesamte Inhalt desselben hierin durch Bezugnahme aufgenommen wird. Bei diesem Beispiel werden die Arraysteuerungsprozesse durch eine Arraysteuerung, die separat von einem Großcomputer ist, durchgeführt, mit dem das dargestellte Plattenarray verbunden ist, während die Dateisystemprozesse durch einen Prozessor durchgeführt werden, der dem Hostcomputer zugeordnet ist.

[0014] Der Ausdruck „Dateisystemzugriffsoperationsanforderung“, wie er in dieser Offenbarung verwendet wird, weist einen computerlesbaren Code auf, der eine Anforderung nach einer bestimmten Dateisystemoperation, die eine Datei einschließt, darstellt. Dateisystemzugriffsoperationen umfassen Operationen zum Erzeugen, Löschen, Öffnen, Schließen, Lesen oder Schreiben einer Datei. Diese Dateisystemzugriffsoperationsanforderungen werden dem Datenspeichersystem gemäß der Erfindung durch Dateisystemklienten kommuniziert, die mit dem System durch eine geeignete Computernetz-kommunikationsanordnung verbunden sind. Dateisystemzugriffsoperationen, die von dem Typ oder der Klasse sind, die gemäß der Erfindung von den Dateisystemprozessen zu den Arraysteuerungsprozessen umgeleitet werden, werden in dieser Offenbarung als „umleitbare“ Operationen bezeichnet. Die Klasse von umleitbaren Operationen kann vorzugsweise durch die Dateigröße definiert sein, obwohl jedes andere Dateiattribut oder jede Kombination von Attributen in dem Bereich der Erfindung verwendet werden kann. Zu Zwecken der Vereinfachung dieser Offenbarung wird angenommen, daß die Dateigröße das Attribut ist, das verwendet wird, um zu definieren, ob eine Operation umleitbar ist oder nicht. Erzeugen-, Löschen-, Lesen- und Schreiben-Operationen, die sich auf eine große Datei beziehen, sind umleitbare Operationen, während Dateizugriffsoperationen, die sich

auf eine kleine Datei beziehen, nicht umleitbare Operationen sind, es sei denn, es ist ein Aufhebungsmerkmal vorhanden, das die Behandlung der Operation unabhängig von der tatsächlichen Dateigröße vorgibt. Es ist ersichtlich, daß das System eine Definition dafür erfordert, was eine „große“ Datei ausmacht. Diese Definition kann einfach eine Schwellendateigröße sein, die in Bytes oder jeder anderen geeigneten Einheit dargestellt ist.

[0015] Das Verfahren gemäß der Erfindung umfaßt zuerst ein Empfangen einer Dateisystemzugriffsoperationsanforderung und ein Bestimmen, ob die Anforderung eine Operation spezifiziert, die eine umleitbare Operation aufweist. Wenn die Dateizugriffsoperationsanforderung eine umleitbare Operation spezifiziert, umfaßt das Verfahren ein Umleiten der umleitbaren Operation von den Dateisystemprozessen zu den Arraysteuerungsprozessen. Das Verfahren umfaßt als nächstes ein Durchführen der umleitbaren Operation mit den Arraysteuerungsprozessen und ein Aktualisieren der Dateisystemdatenverwaltungsanordnung mit Operationsergebnisinformationen von den Arraysteuerungsprozessen.

[0016] Der Dateisystemdatenverwaltungseintrag für jede Datei, die der Gegenstand einer umleitbaren Erzeugen-Operation war, wird von der normalen Struktur, die für Dateien verwendet wird, die der Gegenstand einer nicht umleitbaren Erzeugen-Operation waren, modifiziert. Ein normaler Dateisystemeintrag für eine Datei kann den Dateinamen und verschiedene Dateiattribute gemeinsam mit Informationen hinsichtlich tatsächlicher oder virtueller Speicherraumorte umfassen, die für Daten, die die Datei bilden, zugeordnet sind. Gemäß der vorliegenden Erfindung jedoch schneidet das Dateisystem seinen Datenverwaltungseintrag für eine Datei, die der Gegenstand einer umleitbaren Erzeugen-Operation ist, ab. Insbesondere wird der Eintrag abgeschnitten, um eine Referenz auf die logische Einheit, der die Datei zugeordnet wurde, anstelle von Informationen hinsichtlich Speicherraumorten zu enthalten, die für die Datei durch das Dateisystem zugewiesen sind. Diese Informationen der logischen Einheit werden durch die Arraysteuerungsprozesse als Operationsergebnisinformationen geliefert und sind anstelle der Dateisystemortzuteilungsinformationen für die Datei vorhanden. Dateisystemraumzuteilungsinformationen für die Datei sind nicht verfügbar oder notwendig, da das Dateisystem keinen Dateisystemraum für die Datei zuteilt, wobei Speicherraum statt dessen durch die Arraysteuerungsprozesse zuteilt und verwaltet wird.

[0017] Die Operationsergebnisinformationen, die verwendet werden, um die Dateisystemdatenverwaltungsanordnung für Operationen zu aktualisieren, die direkt durch die Arrayprozesse gehandhabt werden, hängen von der Natur der Operation ab. Für eine umleitbare Erzeugen-Operation umfassen die Operationsergebnisinformationen die Referenz auf die logische Einheit, bei der die Arraysteuerungsprozesse bewirken, daß die Datei gespeichert wird. Wenn die

Operation eine umleitbare Lesen-Operation ist, können die Operationsergebnisinformationen eine Anzeige oder einen Empfang von den Arraysteuerungsprozessen umfassen, um zu zeigen, daß die Lesen-Operation fertiggestellt wurde. Diese Operationsergebnisinformationen werden alle verwendet, um die Dateisystemdatenverwaltungsanordnung zu aktualisieren, so daß alle Funktionen des Dateisystems aufrechterhalten werden, insbesondere einschließlich des Verzeichnisses und der Dateihierarchie.

[0018] Ein Datenspeichersystem gemäß der Erfindung umfaßt ein Datenspeichergerät, das ein Speichermedium aufweist, auf dem Daten gespeichert werden können. Die bevorzugte Form der Erfindung umfaßt tatsächlich ein Array von Massendatenspeichergeräten, wie z. B. Plattenlaufwerken. Das Datenspeichersystem umfaßt auch eines oder mehrere Datenverarbeitungsgeräte oder Datenverarbeitungseinrichtungen. Eine Eingangs/Ausgangs-Anordnung ist mit der Datenverarbeitungseinrichtung zum Weiterleiten von Kommunikationen zwischen einen Dateisystemklienten und der Datenverarbeitungseinrichtung verbunden, während eine Schnittstelle zwischen der Datenverarbeitungseinrichtung und dem Datenspeichergerät Kommunikationen zwischen diesen Elementen des Systems erleichtert. Die Datenverarbeitungseinrichtung führt die Dateisystem- und Arraysteuerungsprozesse aus und führt die oben Bezug nehmend auf das Verfahren gemäß der Erfindung beschriebenen Funktionen durch. Insbesondere bestimmt die Datenverarbeitungseinrichtung, ob eine Dateisystemzugriffsoperationsanforderung eine umleitbare Operation umfaßt. Wenn die Anforderung eine umleitbare Operation umfaßt, wird die Verarbeitungseinrichtung programmiert, um die umleitbare Operation zu den Arraysteuerungsprozessen zu leiten, so daß diese Prozesse die Operation durchführen können. Die Datenverarbeitungseinrichtung ist ebenfalls programmiert, um die Dateisystemdatenverwaltungsanordnung unter Verwendung von Operationsergebnisinformationen für die umleitbare Operation zu aktualisieren.

[0019] Das Datenspeichersystem und das Zugriffsverfahren gemäß der Erfindung liefern alle Vorteile von sowohl Dateisystemen als auch Plattenarrays ohne die normalerweise vorhandenen Nachteile. Insbesondere werden Daten gemäß der gemeinsamen Dateiabstraktion gemäß einem Verzeichnis und einer Dateihierarchie verwaltet. Auf kleine Dateien kann auf die normale Weise mit den Dateisystemprozessen, die mit den Arraysteuerungsprozessen kooperieren, zugegriffen werden. Auf große Dateien kann jedoch direkt durch die Arraysteuerungsprozesse zugegriffen werden, wenn auch der Zugriffsbefehl einen Dateisystembefehl aufweist und die Dateisystemhierarchie beibehalten wird. „Direkter Zugriff“ bedeutet, daß die Arraysteuerungsprozesse eine große Datei von der Speicherung zu dem Dateisystemklienten liefern, ohne daß die Daten für die Datei durch die

Dateisystemprozesse geleitet werden müssen. Dieser direkte Zugriff für große Dateien erhöht die Systemleistung für große Dateien erheblich. Ferner kann das System die gemeinschaftliche Verwendung von sowohl Daten als auch einem Gerät zwischen Dateisystemklienten unterstützen, wobei die Sicherheit wie bei gewöhnlichen Dateisystemen gehandhabt werden kann.

Ausführungsbeispiel

[0020] Diese sowie weitere Ziele, Vorteile und Merkmale der Erfindung werden aus der folgenden Beschreibung der bevorzugten Ausführungsbeispiele Bezug nehmend auf die beigefügten Zeichnungen ersichtlich. Es zeigen:

[0021] **Fig. 1** eine schematische Darstellung eines Plattenarrayspeichersystems des Stands der Technik, das mit einem Hostcomputer gekoppelt ist;

[0022] **Fig. 2** eine schematische Darstellung eines Datenspeichersystems, das die Prinzipien der Erfindung enthält;

[0023] **Fig. 3** eine schematische Darstellung, die Gruppierungen von verschiedenen Softwareprozessen zeigt, die bei dem Datenspeichersystem, das in **Fig. 2** gezeigt ist, operieren;

[0024] **Fig. 4** eine schematische Darstellung, die die Interaktionen zwischen den verschiedenen Softwareprozessen, die in **Fig. 3** gezeigt sind, zeigt;

[0025] **Fig. 5** ein Flußdiagramm, das die Prozessschritte zeigt, die gemäß der Erfindung durchgeführt werden;

[0026] **Fig. 6** eine Darstellung eines Eintrags in die Dateisystemdatenverwaltungsanordnung gemäß der Erfindung; und

[0027] **Fig. 7** eine Darstellung eines Eintrags in eine Arraysteuerungsdatenverwaltungsanordnung gemäß der Erfindung.

[0028] Bezug nehmend auf **Fig. 2** umfaßt ein Datenspeichersystem **201**, das die Prinzipien der Erfindung aufweist, eine Datenverarbeitungsanordnung **202** und ein Array **203a** separater Speichergeräte **203**. In der Form der Erfindung, die in **Fig. 2** gezeigt ist, umfaßt die Datenverarbeitungsanordnung **202** zwei separate Prozessoren, eine Arraysteuerung **204** und einen Dateisystemprozessor **205**. Ein Systembus **206** verbindet die beiden Prozessoren **204** und **205**, einen Speichergerätadapter **207** und ein Eingangs/Ausgangs-Tor **208**. Der separate Adapter **207**, der in **Fig. 2** gezeigt ist, liefert eine Schnittstelle mit dem Speichergerätsbus **209**. Außerdem ist ein Hardwareprotokollstapel **210** dem Eingangs/Ausgangs-Tor **208** zum Liefern von Netzkommunikationsdiensten zugeordnet, die es dem System **201** ermöglichen, mit einem Computernetz **211** zu kommunizieren. Das Computernetz **211** umfaßt einen oder mehrere Dateisystemklienten **212**, die einen Dateisystemzugriff auf das Speichersystem **201** haben. Diese Dateisystemklienten **212** stellen separate Rechengерäte dar, wie z. B. Personalcomputer oder Ar-

beitsplatzrechner auf dem Netz **211**.

[0029] Die einzelnen Speichergeräte **203**, die in **Fig. 2** gezeigt sind, weisen vorzugsweise Plattenlaufwerke oder andere geeignete Massenspeichergeräte auf. Obwohl dies in **Fig. 2** nicht gezeigt ist, ist zu erkennen, daß diese Speichergeräte **203** jeweils ein Schnittstellenverbindungselement und eine Treibersteuerung umfassen. Das Schnittstellenverbindungselement ermöglicht es, daß das jeweilige Gerät mit dem Speichergerätsbus **209** verbunden ist, wobei die Treibersteuerung ein Verarbeitungselement zum Ausführen eines Programmcodes aufweist, um die Operation des bestimmten Speichergeräts zu steuern.

[0030] Das dargestellte Speichersystem **201** kann in einem Gehäuse (nicht gezeigt) untergebracht sein, das die Datenverarbeitungsanordnung **202**, das Eingangs/Ausgangs-Tor **208** und die Protokollstapelhardware **210**, den Speichergerätschnittstellenadapter **207** und eines oder mehrere Speichergeräte **203** unterbringt. Der dargestellte Speichergerätsbus **209** umfaßt ein externes Verbindungselement **213** zum Verbinden zusätzlicher Speichergeräte außerhalb des Systemgehäuses.

[0031] Die bestimmte Struktur des Speichersystems **201**, das in **Fig. 2** dargestellt ist, ist nur als eine bevorzugte Form der Erfindung gezeigt. Viele Variationen bezüglich der Struktur sind in dem Bereich der Erfindung möglich. Obwohl z. B. zwei separate Datenverarbeitungsgeräte **204** und **205** in **Fig. 2** gezeigt sind, ist es ersichtlich, daß alle Datenverarbeitungsfunktionen, die bei der Erfindung erforderlich sind, auf einem einzelnen Datenverarbeitungsgerät durchgeführt werden oder zwischen mehr als zwei Datenverarbeitungsgeräten aufgeteilt sein können. Außerdem können die Funktionen, die durch den Speichergerätschnittstellenadapter **207**, der in **Fig. 2** gezeigt ist, geliefert werden, durch ein Verarbeitungsgerät, das in der Verarbeitungseinrichtung **202** enthalten ist, durchgeführt werden. Die Funktionen, die durch den Protokollstapel **210**, der in **Fig. 2** gezeigt ist, geliefert werden, können ebenso durch eine andere Verarbeitungseinrichtung in dem System, wie z. B. die Verarbeitungseinrichtung **202** durchgeführt werden. Ferner ist, obwohl ein externes Speichergerät **203** und vier interne Speichergeräte **203** zu Beispielzwecken in **Fig. 2** gezeigt sind, die Erfindung nicht auf diese oder eine andere Zahl von Speichergeräten beschränkt. Obwohl die Erfindung auf keinen bestimmten Typ von Speichergeräten beschränkt ist, können die Geräte SCSI-Geräte sein, wobei in diesem Fall der Bus **209** einen SCSI-Bus aufweisen würde und der Adapter **207** einen SCSI-Adapter. Der Systembus **206** kann jeden geeigneten Busstandard, wie z. B. PCI, aufweisen.

[0032] Für Fachleute ist es ebenfalls ersichtlich, daß es möglich ist, die vorliegende Erfindung in Verbindung mit einem Hostcomputer zu implementieren, und nicht als eine alleinstehende Netzeinheit, die in **Fig. 2** dargestellt ist. Bei dieser alternativen Imple-

mentierung können der Arraysteuerungsprozessor und der Speichergerätadapter auf einer hinzufügbaren Karte, wie z. B. einer PCI-Karte, implementiert sein, die mit dem PCI-Bus eines Hostcomputers zu verbinden ist, wobei die Speichergeräte alle außerhalb des Hostcomputers sein könnten. Die Prozesse, die durch den Dateisystemprozessor **205** in **Fig. 2** durchgeführt werden, würden bei dieser alternativen Form der Erfindung durch einen Prozessor des Hostcomputers durchgeführt werden, wobei das Eingangs/Ausgangs-Tor und der Protokollstapel des Hostcomputers für Netzkommunikationen verwendet werden könnten.

[0033] **Fig. 3** zeigt weitere Details der Programmprozesse, die durch die Datenverarbeitungsanordnung **202** unter der Steuerung eines Computerprogrammcodes gemäß der Erfindung durchgeführt werden. Die Arraysteuerung **204** führt eine Zahl von Programmprozessen durch oder führt dieselben aus, wobei jeder derselben in einem separaten Kasten gezeigt ist, die in dem Kasten **204** in **Fig. 3** enthalten sind. Diese Arraysteuerungsprogrammprozesse werden durch einen Arraysteuerungsprogrammcode implementiert und werden in dieser Offenbarung kollektiv als Arraysteuerungsprozesse **300** bezeichnet. In der dargestellten Form der Erfindung umfassen die Arraysteuerungsprozesse **300** einen Arraybefehlsinterpretierer **301**, eine virtuelle Arrayabbildungskomponente **302** und eine RAID-Verwaltungskomponente **303**. Diese Elemente stellen jeweils einen Computerprogrammcode dar, der durch die Arraysteuerung **204** ausgeführt wird, um verschiedene Arraysteuerungsfunktionen durchzuführen. **Fig. 3** zeigt auch einen Array-Cache-Speicher **304**, der dem Arraysteuerungsprozessor **204** zugeordnet ist. Dieser Array-Cache-Speicher kann einen Direktzugriffsspeicher aufweisen, der der Arraysteuerung zugeordnet ist, oder kann einen batteriegepufferten Direktzugriffsspeicher aufweisen. Der Speicher wird verwendet, um Daten, die unter Steuerung der Arraysteuerungsprozesse **300** zu und von den Speichergeräten übertragen werden, einer Cache-Speicherung zu unterziehen.

[0034] Der Arraybefehlsinterpretierer **301**, der in **Fig. 3** gezeigt ist, ist für ein Empfangen eines auf das Datenspeichergerät bezogenen Befehls und für ein Umwandeln des Befehls in Instruktionen oder Rufe nach anderen Arraysteuerungsprozessen verantwortlich. Bei der bevorzugten Form der Erfindung umfaßt der Arraybefehlsinterpretierer **301** auch einen Bytezugriffsumwandlungsprogrammcode zum Umwandeln der Byteversatzzugriffsinstruktion, die aus einer umleitbaren Operation resultiert, in eine Blockzugriffsinstruktion, wie unten genauer erläutert wird. Eine virtuelle Arrayabbildungskomponente **302** liefert eine Arraydatenverwaltungsanordnung, die Daten, die auf den Speichergeräten gespeichert sind, auf logische Einheiten und auf Blöcke von Speicherraum bezieht. Ein Eintrag in diese Arraydatenverwaltungsanordnung ist in **Fig. 7**, die unten beschrieben wird, gezeigt. Die virtuelle Abbildungskomponente **302** ist

vorzugsweise auch für ein Erzeugen und Zuweisen von logischen Einheiten für die Daten, die über das Gerätarray **203a** gespeichert werden sollen, verantwortlich. Eine RAID-Komponente **303** ist für eine Verwaltung des Speichergeräatarrays **203a** und für Kommunikationen mit der virtuellen Arrayabbildungskomponente verantwortlich. Die RAID-Komponente **303** implementiert auch alle Datenspeicherredundanzschemata oder RAID-Schemata, die durch das Speichersystem **201** unterstützt werden. Die Arraysteuerungsprozesse sind in der Technik von Plattenarrays und RAID-Systemen bekannt und werden hier nicht weiter beschrieben, um die Erfindung nicht mit unnötigen Details zu verschleiern.

[0035] Der Dateisystemprozessor **205**, der in den **Fig. 2** und **3** gezeigt ist, führt eine Zahl von Programmprozessen durch oder führt dieselben aus, wobei jeder derselben durch einen Computerprogrammcode implementiert ist. Jeder Prozeß ist in einem separaten Kasten gezeigt, die in dem Kasten **205**, der in **Fig. 3** gezeigt ist, enthalten sind. Zusätzlich zu den Dateisystemprozessen oder dem Dateisystem **308**, das durch einen Dateisystemprogrammcode implementiert ist, führt der Programmsystemprozessor **205** eine Plattentreiberkomponente **309**, eine Keil- oder Wedge-Dateisystemkomponente **310** und eine Bytemodustreiberkomponente **311** aus. Jede dieser Komponenten stellt Programmprozesse dar, die durch einen Computerprogrammcode implementiert werden, der durch den Dateisystemprozessor **205** ausgeführt wird. **Fig. 3** zeigt auch einen Dateisystem-Cache-Speicher **312**, der einen Direktzugriffsspeicher aufweist, der den Dateisystemprozessor **205** zugeordnet ist. Der Dateisystem-Cache-Speicher **312** liefert Speicherraum zum Cache-Speichern von Dateidaten, die letztendlich zu und von Datenspeichergeräten **203**, die in **Fig. 2** gezeigt sind, kommuniziert werden.

[0036] Es ist ersichtlich, daß die Dateisystemprozesse, die in **Fig. 3** bei Block **308** gezeigt sind, alle Programmprozesse umfassen, die zum Implementieren des erwünschten Dateisystems notwendig sind. Insbesondere liefern Dateisystemprozesse **308** eine Dateisystemdatenverwaltungsanordnung. Die Dateisystemdatenverwaltungsanordnung umfaßt eine oder mehrere Datentabellen, die jede Datei in dem Dateisystem auf verschiedene Dateiattribute beziehen. Außerdem beziehen, wie später erklärt wird, die Dateisystemdatenverwaltungstabellen jede Datei auf entweder Speichergeräzuteilungsinformationen oder eine Referenz auf eine logische Einheit, die durch Arraysteuerungsprozesse **300** erzeugt und verwaltet wird. Für Kommunikationen mit den Arraysteuerungsprozessen **300** beruhen Dateisystemprozesse **308** auf einer Plattentreiberkomponente **309**. Die Plattentreiberkomponente **309** umfaßt einen Computerprogrammcode zum Kommunizieren von Dateisystemdatenzugriffsbefehlen von den Dateisystemprozessen **308** zu den Arraysteuerungsprozessen **300**, um auf bestimmte Dateisystemdaten, die an

den Datenspeichergeräten **203** gespeichert sind, zuzugreifen. Insbesondere liefert die Plattentreiberkomponente **309** jede notwendige Codierung von Instruktionen und Daten, die zwischen den Dateisystemprozessen **308** und den Arraysteuerungsprozessen kommuniziert werden. Es ist ersichtlich, daß die Plattentreiberkomponente betrachtet werden kann, um in den Dateisystemprozessen **308** enthalten zu sein, bei dieser Offenbarung jedoch separat dargestellt ist, um ihre bestimmte Funktion besser zu beschreiben. [0037] Wie bei den meisten Arraysteuerungsprozessen **300** sind die spezifischen Implementierungen und die Operation der Dateisystemprozesse **308** und der Plattentreiberkomponente **309** Fachleuten auf diesem Gebiet bekannt. Deshalb werden Implementierungs- und Operationsdetails in dieser Offenbarung nur wie notwendig beschrieben, um die vorliegende Erfindung klar und vollständig zu offenbaren. [0038] Die Keildateisystemkomponente **310** implementiert spezielle Dateisystemprozesse, die zwischen Dateisystemklienten (**212** in **Fig. 2**) und Dateisystemprozesse **308** eingefügt sind. Wie weiter unten Bezug nehmend auf **Fig. 4** und **5** beschrieben wird, empfängt die Keildateisystemkomponente **310** Dateisystemzugriffsoperationsanforderungen von Dateisystemklienten und bestimmt, ob die Anforderung eine umleitbare Operation umfaßt. Die Bestimmung wird durch einen Operationserfassungsprogrammcode, der in dem Keildateisystem enthalten ist, durchgeführt. Die Keildateisystemkomponente **310** leitet nicht umleitbare Operationen zu Dateisystemprozessen **308** weiter. Diese Verarbeitung nicht umleitbarer Operationen durch die Keilsystemprozesse **308** tritt wie bei einem Dateisystem des Stands der Technik auf. Wenn die Keildateisystemkomponente **310** jedoch eine umleitbare Operation erfaßt, verhindert dieselbe, daß die Operation die Keilsystemprozesse **308** erreicht, und leitet die umleitbare Operation statt dessen um, um durch die Arraysteuerungsprozesse **300** verarbeitet zu werden, die bei der Arraysteuerung **204** ausgeführt werden. Weitere Details der Operation der Keildateisystemkomponente **310** werden unten Bezug nehmend auf die **Fig. 4** und **5** beschrieben.

[0039] Eine Verwendung der Keildateisystemkomponente **310**, die in **Fig. 3** gezeigt ist, um Dateisystemzugriffsoperationsanforderungen abzufangen, weist den Vorteil auf, daß die Operationserfassungsfunktion implementiert sein kann, ohne Veränderungen an dem Programmcode, der Dateisystemprozesse **308** implementiert, vorzunehmen. Es ist jedoch ersichtlich, daß die oben Bezug nehmend auf die Keildateisystemkomponente **310** beschriebene Erfassungsfunktion der umleitbaren Operation alternativ durch ein Modifizieren von Dateisystemprozessen **308** geliefert werden kann. Bei dieser alternativen Implementierung würden umleitbare Operationen Dateisystemprozesse **308** erreichen, jedoch nicht verarbeitet und statt dessen zu Arraysteuerungsprozessen **300** umgeleitet werden. Diese alter-

native Implementierung kann als ein Äquivalent zu der Keildateisystemkomponentenimplementierung betrachtet werden, die in dieser Offenbarung detailliert dargelegt ist.

[0040] Eine Bytemodustreiberkomponente **311** weist einen Programmcode zum Überbringen von Befehlen für eine umleitbare Zugriffsoperation zu Arraysteuerungsprozessen **300** auf, und insbesondere zu der Arraybefehlsinterpretiererkomponente **301**. Die Überbringungsfunktionen, die durch die Bytemodustreiberkomponenten **311** für umleitbare Operationen durchgeführt werden, sind analog zu den Funktionen, die durch die Plattentreiberkomponente **309** hinsichtlich nicht umleitbarer Operationen durchgeführt werden.

[0041] Wie in **Fig. 3** gezeigt ist, kommunizieren die Programmprozesse, die bei der Arraysteuerung **204** und dem Dateisystemprozessor **205** durchgeführt werden, miteinander, wobei auch beide mit Dateisystemklientenanwendungen **314** kommunizieren, die Software aufweisen, die bei Dateisystemklienten **212** in **Fig. 2** ausgeführt wird. Die Arraysteuerung **204** kommuniziert auch mit Plattensteuerungsprozessen **316**. Die Plattensteuerungsprozesse **316** weisen einen Programmcode auf, der an der Steuerung (nicht gezeigt), die jedem Datenspeichergerät **203** in **Fig. 2** zugeordnet ist, ausgeführt wird.

[0042] Das Verfahren der Erfindung und die Operation der verschiedenen Programmprozesse können Bezug nehmend auf die **Fig. 4** und **5** beschrieben werden. Wie bei einem Prozeßblock **501** in **Fig. 5** gezeigt ist, umfaßt das Verfahren ein Empfangen einer Dateisystemoperationsanforderung.

Diese Dateisystemzugriffsoperationsanforderung wird von einem Dateisystemklienten **212** der **Fig. 2** und **4** durch Kommunikationseinrichtungen, die durch den Protokollstapel **210** geliefert werden, der ebenfalls in diesen Figuren gezeigt ist, empfangen. Die Zugriffsoperationsanforderung wird bei der bevorzugten Form der Erfindung durch die Keildateisystemkomponente **310** empfangen. Wie oben angemerkt wurde, ist diese Keildateisystemkomponente **310** zwischen den Dateisystemprozessen **308** und Dateisystemklienten **212** eingefügt. Eine Komponente, die als eine virtuelle Dateisystemschnittstelle bezeichnet wird, kann bei bestimmten Dateisystemen verwendet werden, um Dateisystemmodifizierungen vor den Dateisystemkomponenten einzufügen, oder um ein zusätzliches oder alternatives Dateisystem einzufügen. So müssen die Dateisystemprozesse **308**, die gemäß der bevorzugten Form der Erfindung, die in den Figuren dargestellt ist, verwendet werden, eine virtuelle Dateisystemschnittstelle oder ein anderes Verfahren zum Einfügen der Keildateisystemkomponente **310** unterstützen.

[0043] Als nächstes umfaßt das Verfahren den Schritt des Bestimmens, ob die Dateisystemzugriffsoperationsanforderung eine umleitbare Operation umfaßt. Dieser Schritt ist bei einem Entscheidungsblock **502** in **Fig. 5** gezeigt und wird bei der darge-

stellten Form der Erfindung durch die Keildateisystemkomponente **310** durchgeführt. Wie oben erläutert wurde, wird zu Zwecken dieser Offenbarung angenommen, daß Dateisystemzugriffsoperationen allgemein basierend auf der Größe der Datei (Datenmenge, die die Datei bildet), die der Zugriffsoperationsanforderung unterzogen wird, als umleitbare oder nicht umleitbare Operationen definiert sind. Diese Umleitung basierend auf der Dateigröße weist den Effekt auf, daß sie bewirkt, daß alle Dateien über der vorbestimmten Schwellengröße direkt durch Arraysteuerungsprozesse **300** gehandhabt oder verarbeitet werden, und daß sie bewirkt, daß alle Dateien unter der Schwellengröße normal durch Dateisystemprozesse **308** gehandhabt oder verarbeitet werden. Die bevorzugte Form der Erfindung umfaßt auch einen Mechanismus zum Aufheben der normalen Regeln zum Bestimmen, ob eine Operation eine umleitbare Operation ist. Wieder können, obwohl die Erfindung hier zu Beispielzwecken beschrieben wird, um die Dateigröße zu verwenden, um zu bestimmen, ob eine Operation umleitbar ist, andere Dateiattribute oder andere Standards oder Kombinationen von Attributen und Standards in dem Bereich der angefügten Ansprüche beim Definieren von Zugriffsoperationen als umleitbare oder nicht umleitbare Operationen verwendet werden.

[0044] Jedes geeignete Verfahren kann zum Durchführen der Bestimmung an dem Entscheidungsblock **502** verwendet werden. Insbesondere kann bei dem Beispiel, bei dem die Bestimmung basierend auf der Größe der Datei durchgeführt wird, die der empfangenen Operation unterzogen wird, die Keildateisystemkomponente **310** Dateigrößeninformationen von der Dateisystemdatenverwaltungsanordnung erhalten, die den Dateisystemprozessen **308** zugeordnet ist. So kann der Entscheidungsschritt bei Block **502** ein Senden einer Anforderung nach den notwendigen Dateigrößeninformationen von der Keildateisystemkomponente **310** zu den Dateisystemprozessen **308** umfassen. Alternativ kann es vorzuziehen oder notwendig sein, daß das Keildateisystem die notwendigen Dateigrößeninformationen von der ankommenden Dateizugriffsoperationsanforderung erhält, wobei in diesem Fall der Entscheidungsschritt bei Block **502** ein Lesen von Dateigrößeninformationen von der ankommenden Operationsanforderung umfassen würde. Wenn jedoch keine Dateigrößeninformation für die Datei verfügbar sind, die der ankommenden Dateizugriffsoperationsanforderung unterzogen wird, kann es notwendig sein, daß die Keildateisystemkomponente **310** programmiert wird, um eine Annahme hinsichtlich der Größe der Datei zu machen. Es kann alternativ möglich sein, daß die Keildateisystemkomponente **310** Dateigrößeninformationen von dem Dateisystemklienten **202**, der die Anforderung ausgibt, erhält, obwohl gegenwärtige Dateisysteme eine derartige Funktion nicht unterstützen.

[0045] Alternativ können einige bevorzugte Formen der Erfindung jeder Datei ein spezielles Attribut zu-

ordnen. Bei dem vorliegenden Beispiel, bei dem die Subjektdateigröße beim Bestimmen verwendet wird, ob eine Operation umleitbar ist, kann das Dateisystem modifiziert sein, um ein Dateistatusattribut als entweder „groß“ oder „klein“ einzustellen, wobei jedes Attribut in der Dateisystemdatenverwaltungsanordnung, die durch Dateisystemprozesse **308** implementiert ist, gespeichert sein kann. Der Schritt des Bestimmens, ob die Operation eine umleitbare Operation ist, weist dann einfach ein Lesen des Dateistatusattributs, das bei der Dateisystemdatenverwaltungsanordnung für die Datei gespeichert ist, die der Zugriffsoperation unterzogen wird, auf. Dieses Dateistatusattribut liefert auch ein bequemes Verfahren zum Aufheben der Standardregeln. Dies bedeutet, daß ein Aufhebungsmerkmal des Keildateisystems bewirken kann, daß das Dateistatusattribut unabhängig von der tatsächlichen Dateigröße als der Aufhebungsstatus gespeichert wird. Andernfalls kann ein anderer Code oder ein anderes Flag bei der Keildateisystemkomponente **310** verwendet werden, um sicherzustellen, daß eine bestimmte Operation unabhängig von den tatsächlichen Dateiattributen als umleitbar oder nicht umleitbar behandelt wird.

[0046] Es sei darauf hingewiesen, daß bestimmte Dateizugriffsoperationen unabhängig von den Dateiattributen der Datei, die der Dateizugriffsoperationsanforderung unterzogen wird, keine umleitbaren Operationen sein können. Insbesondere können die Datei-Öffnen- und Schließen-Operationen an dem Dateisystempegel keine umleitbaren Operationen sein, da diese Operationen kein Lesen oder Schreiben von Daten für die Subjektdatei zu den Speichergeräten **203** umfassen sowie keine Speicherraumteilungen beinhalten. So gibt es keinen Grund, eine Öffnen- oder Schließenoperation um die Dateisystemprozesse **308** herumzuleiten.

[0047] Wenn bestimmt wird, daß die Operation eine umleitbare Operation ist, leitet das Keildateisystem **310** die Operation zu Arraysteuerungsprozessen **300** um, wie bei einem Block **503** in **Fig. 5** gezeigt ist. Bei der bevorzugten Form der Erfindung umfaßt das Verfahren auch ein Umwandeln der umgeleiteten Dateisystemoperation von einer Byteversatzdefinition zu einer Blockzugriffsdefinition. Dieser Umwandlungsschritt ist bei einem Prozeßblock **504** in **Fig. 5** gezeigt. Diese Umwandlung von einem Byteversatz und einer Bytelänge zu einem Blockversatz und einer Blocklänge wird vorzugsweise durch einen Programmcode unter Verwendung eines Speichers, der bei der Arraysteuerung **204** verfügbar ist, durchgeführt, der in dem Arraybefehlsinterpretierer **301** enthalten oder demselben zugeordnet ist. Es sei darauf verwiesen, daß eine ähnliche Umwandlung Teil der normalen Operation der Dateisystemprozesse **308** ist. In jedem Fall umfaßt die Umwandlung ein Teilen der Byteversatz- und Längendaten durch die Zahl von Bytes in jedem Block von Speicherraum, der für Speichergeräte **203** definiert ist. Blöcke sind üblicherweise als 512-Byte-Speicherräume definiert, obwohl

die Erfindung nicht auf diese Blockgröße beschränkt ist.

[0048] Sobald die umgeleitete Dateisystemoperation bei dem Arraybefehlsinterpretierer **301** zu einer Blockzugriffsform umgewandelt ist, wird die Operation durchgeführt, wie bei einem Schritt 505 in **Fig. 5** gezeigt ist. Ein Durchführen der angeforderten Operation umfaßt allgemein die normalen Schritte bei einem Arrayzugriff. Insbesondere nimmt der Arraybefehlsinterpretierer **301** den Code oder die Instruktion, die die umgeleitete Operation darstellt, und gibt abhängig von der Natur der Operation Rufe oder Befehle an die zugeordneten Prozesse aus. Für eine Lesen-Operation bestimmt die Arrayabbildungskomponente **302** die Blockzuteilungen, die der vorher existierenden Datei, die gelesen werden soll, zugeteilt sind. Die RAID-Verwaltungskomponente **303** gibt dann einen geeigneten Befehl an das Gerät oder Geräte **203** aus, um die Daten, die an den identifizierten Blöcken gespeichert sind, zu lesen. Bei einer Schreiben-Operation bildet die Abbildungskomponente **302** die Dateidaten auf Blöcke in der logischen Einheit ab, die der Datei zugewiesen sind. Die RAID-Verwaltungskomponente **403** gibt dann einen Befehl an das Speichergerät oder -Geräte **203** aus, um die empfangenen Daten zu den abgebildeten Blöcken zu schreiben.

[0049] Es gibt verglichen mit Lesen- und Schreiben-Operationen an ein Speicherarray, die bei Systemen des Stands der Technik durchgeführt werden, bei der Leistung von umleitbaren Lesen- und Schreiben-Operationen einen merklichen Unterschied. Bei der bevorzugten Form der Erfindung werden Daten für die Lesen- und Schreiben-Operationen direkt zwischen den Arraysteuerungsprozessen **300** und den Dateisystemklienten (**212** in **Fig. 2** und **4**) übertragen, wobei die Dateisystemprozesse **308** vollständig umgangen werden. Der Dateiumgehungspfad wird in dem folgenden Absatz Bezug nehmend auf **Fig. 4** erläutert. Hinsichtlich der umleitbaren Lesen-Operationen, bei denen Dateidaten wie durch den Dateisystemklienten **212** angeleitet zurückgegeben werden, erfordert diese Umgehung der Dateisystemprozesse, daß die Arraysteuerungsprozesse **300** angepaßt sind, um die Datenmeldung gemäß den Standards des Dateisystems zu formatieren. Diese Formatierung der Daten, die ansprechend auf eine umleitbare Lesen-Operation von Geräten **203** gelesen werden, wird vorzugsweise durch einen Dateisystemformatierungsprogrammcode durchgeführt, der in dem Arraybefehlsinterpretierer **301** enthalten ist.

[0050] **Fig. 4** zeigt drei separate Pfade zu dem Arraybefehlsinterpretierer **301**. Ein Pfad **401** läuft von dem Protokollstapel **210** zu dem Interpretierer **301**. Dieser Pfad **401** ist gezeigt, um anzuzeigen, daß die Dateidaten, die einer umleitbaren Operation zugeordnet sind, Dateisystemprozesse vollständig umgehen, die entweder zu oder von den Speichergeräten **203** verlaufen. Ein Pfad **402** zeigt einen Pfad zwischen der Keildateisystemkomponente **310** und dem

Interpretierer **301**, um anzuzeigen, daß Befehle und Empfänger, die umleitbaren Operationen zugeordnet sind, zwischen diesen beiden Prozessen verlaufen. Ein Pfad **403** zwischen der Plattentreiberkomponente **309** und dem Arraybefehlsinterpretierer **301** ist in **Fig. 4** gezeigt, um anzuzeigen, daß Befehle, Empfänger und Dateidaten, die einer nicht umleitbaren Dateisystemzugriffsoperation zugeordnet sind, zwischen dem Plattentreiber **309** und dem Arraybefehlsinterpretierer **301** verlaufen, wie weiter unten erläutert wird.

[0051] Wie bei einem Prozeßblock **506** in **Fig. 5** gezeigt ist, umfaßt das Verfahren auch den Schritt des Aktualisierens der Dateisystemdatenverwaltungsanordnung mit Informationen über die Ergebnisse der ausgeführten, umleitbaren Zugriffsoperation. Da die Erfindung die erwünschte Dateihierarchie durch die Dateisystemdatenverwaltungsanordnung beibehält, müssen die Tabellen, die die Anordnung ausmachen, aktualisiert sein, um die korrekten Informationen hinsichtlich der Datei beizubehalten, die einer umleitbaren Zugriffsoperation unterzogen wird, die direkt durch Arraysteuerungsprozesse **300** ausgeführt wird. Arraysteuerungsprozesse **300** erleichtern diese Aktualisierung der Dateisystemdatenverwaltungsanordnung durch ein Kommunizieren von Operationsergebnisinformationen zu den Dateisystemprozessen **308**. Die Natur der Operationsergebnisinformationen hängt von der Natur der umleitbaren Zugriffsoperation ab, die durchgeführt wurde. Für eine Erzeugen-Operation, die zu Arraysteuerungsprozessen **300** umgeleitet und von denselben durchgeführt wird, umfassen die Ergebnisinformationen eine Referenz auf eine logische Einheit, die für die Datei durch die Arraysteuerungsprozesse zugewiesen wird, insbesondere die Arrayabbildungskomponente **302**. Für eine umleitbare Lesen- oder Schreiben-Operation können die Operationsergebnisinformationen einen Empfang oder eine Meldung aufweisen, die die Fertigstellung der bestimmten Operation anzeigt. In jedem Fall können die Operationsergebnisinformationen direkt zurück zu Dateisystemprozessen **308** kommuniziert werden oder durch die Keildateisystemkomponente **310** kommuniziert werden.

[0052] Wieder Bezug nehmend auf den Entscheidungsblock **502** leitet die Keildateisystemkomponente **310**, wenn bestimmt wird, daß die Zugriffsoperationsanforderung eine Operation umfaßt, die keine umleitbare Operation ist, die Operation direkt zu Dateisystemprozessen **308**, wie bei einem Prozeßblock **508** angezeigt ist. Dateisystemprozesse **308** führen dann die Operation bei einem Block **509** wie bei Systemen des Stands der Technik durch. Bei einer Lesen-Operation z. B. wird die Dateisystemdatenverwaltungsanordnung befragt, um die Orte, die der Datei zugeteilt sind, zu bestimmen. Dateisystemprozesse **308** bauen dann den geeigneten Befehl auf und geben denselben aus, wobei die Plattentreiberkomponente **309** den Befehl codiert und dem Arraybefehlsinterpretierer **301** überbringt. Es wird darauf

hingewiesen, daß die Umwandlung von einem Bytezugriff zu einem Blockzugriff in diesem Fall durch Dateisystemprozesse **308** durchgeführt wird. Die Prozesse an der Arraysteuerung **204** antworten auf die Kommunikation von der Plattentreiberkomponente **309**, indem sie bewirken, daß die Daten von den spezifizierten Blöcken in der logischen Einheit, die für das Dateisystem zugeteilt ist, gelesen werden. Die Daten werden an sowohl dem Array-Cache-Speicher **304** als auch dem Dateisystem-Cache-Speicher **312** Cache-gespeichert und dann wie durch die Dateisystem-Lesen-Operation-Anforderung, die den Zugriff fordert, angeleitet zurückgegeben.

[0053] In dem Fall einer nicht umleitbaren Schreiben-Operation aktualisieren Dateisystemprozesse **308** die Dateisystemdatenverwaltungsanordnung für die virtuellen Speicherräume, die den Dateidaten zugeteilt sind. Die Dateisystemprozesse **308** verwenden dann die Plattentreiberkomponente **309**, um mit Arraysteuerungsprozessen **300** zu kommunizieren, um die Dateidaten und die Dateisystemraumzuteilungsinformationen zu liefern, die zum Schreiben der Daten erforderlich sind. Die Daten werden an die spezifizierten Blöcke geschrieben, die für die Datei in der logischen Einheit zugeteilt sind, die für das Dateisystem zugewiesen ist.

[0054] Es wird darauf hingewiesen, daß gemäß der bevorzugten Form der Erfindung eine einzelne logische Einheit der Plattenarrayimplementierung dem Speichern kleiner Dateien gewidmet ist, die durch Dateisystemprozesse **308** auf die normale Weise verarbeitet werden. Alle großen Dateien (bei diesem Beispiel, bei dem die Dateigröße die Klasse von umleitbaren Operationen bestimmt) werden jedoch direkt durch Arraysteuerungsprozesse, wie durch den Pfad **401** in **Fig. 4** angezeigt ist, verarbeitet und vorzugsweise jeweils ihrer logischen Einheit durch Arraysteuerungsprozesse **300** zugeteilt.

[0055] Das Ausführungsbeispiel der Erfindung, das in den Figuren gezeigt ist, zeigt das Dateisystem, das ohne die Hilfe einer Volumenverwaltersoftwarekomponente mit den Arraysteuerungsprozessen **300** in Wechselwirkung steht. Für Fachleute ist es jedoch ersichtlich, daß die Erfindung unter Verwendung von Volumenverwaltersoftwareprozessen implementiert sein kann, die zwischen dem Dateisystem **308** und Arraysteuerungsprozessen **300** eingefügt sind. Der Volumenverwalter würde funktionieren, um zu bewirken, daß mehrere logische Einheiten des Plattenarrays für das Dateisystem **308** als eine einzelne logische Einheit erscheinen.

[0056] Bei einer Dateisystemimplementierung geht eine Öffnungsoperation einer Lesen- oder Schreiben-Operation voraus. Die separate Öffnungsoperation wird verwendet, um Konflikte zwischen Dateisystemklienten **212** (**Fig. 2** und **4**), die auf eine gemeinsame Datei zugreifen, zu verhindern. Das Dateisystem, wie z. B. das System **308**, erzeugt auch eine temporäre Datenstruktur zur Verwendung beim Zugreifen auf Dateien. Unter anderen Funktionen be-

zieht diese temporäre Datenstruktur jede Datei auf eine Kennung oder einen anderen Identifizierer und speichert zusätzliche Dateiverwaltungsinformationen von der Dateisystemdatenverwaltungsanordnung, um ein Verarbeiten nachfolgender Lesen- und Schreiben-Operationen zu beschleunigen.

[0057] Bei der bevorzugten Form der Erfindung umfaßt das Keildateisystem **310** einen temporären Datenstrukturprogrammcode, der eine ähnliche temporäre Datenstruktur für Dateien erzeugt und verwaltet, die der Gegenstand umleitbarer Operationen sein können. Jede Aufzeichnung in dieser temporären Dateistruktur des Keildateisystems umfaßt vorzugsweise eine Dateikennung, den Dateinamen und einen Identifizierer der logischen Einheit. Der Identifizierer der logischen Einheit weist einen Identifizierer für die logische Einheit auf, bei der die jeweilige Datei gemäß einer umleitbaren Operation gespeichert wurde. Diese temporäre Dateistruktur des Keildateisystems hilft die Verarbeitung einer umleitbaren Operation für eine spezifizierte Datei zu beschleunigen, indem erforderliche Zugriffsinformationen für die spezifizierte Datei zu dem Zeitpunkt, zu dem die umleitbare Operation empfangen wird, ohne weiteres verfügbar sind.

[0058] Die **Fig. 6** und **7** zeigen Teile der Datenverwaltungsanordnungen, die durch die Dateisystemprozesse **308** bzw. die Arraysteuerungsprozesse **305** verwendet werden. Bezug nehmend auf **Fig. 6** ist bei **601** ein Dateisystemdatenverwaltungseintrag gezeigt. Bei dieser bevorzugten Form der Erfindung umfaßt jeder Eintrag ein Dateiname-Feld **603**, eine Gruppe **604** von Dateiattributfeldern und ein Feld **605**, das Informationen hinsichtlich einer Dateisystemspeicherraumzuteilung enthält. Das Attributfeld **606** speichert bei der bevorzugten Form der Erfindung ein Dateistatusattribut. Dieses Dateistatusattribut definiert die Datei als entweder eine kleine Datei oder eine große Datei. Ein Attributfeld **607** liefert einen Ort zum Speichern einer Referenz auf eine logische Einheit, bei der die Subjektdateteil gespeichert wird. Für kleine Dateien gibt es keinen Wert in diesem Attributfeld **607** der logischen Einheit, da Arraysteuerungsprozesse **300** kleine Dateien nicht direkt verwalten und kleinen Dateien keine eigenen logischen Einheiten zuweisen. Vielmehr werden kleine Dateien gemäß der Erfindung durch Dateisystemprozesse **308** gehandhabt. Dateisystemprozesse **308** teilen Speicherraum für die Datei in einer einzelnen logischen Einheit zu, die für das Dateisystem durch Arraysteuerungsprozesse **300** zugewiesen ist. Diese Zuteilungsinformationen werden für eine kleine Datei in Feld **605** gespeichert. Im Gegensatz dazu wird das Dateisystemraumzuteilungsfeld **605** vorzugsweise bei einem Eintrag einer großen Datei leer gelassen, während das Referenzattributfeld **607** der logischen Einheit die Referenz auf die logische Einheit, bei der die große Datei gespeichert ist, durch die Arraysteuerungsprozesse **300** speichert. Das Raumzuteilungsfeld **605** kann für eine große Datei leer gelassen wer-

den, da ein Speicherraum für die Datei zugeteilt und durch die Arraysteuerungsprozesse **300** verwaltet wird. So ist es nicht notwendig, daß die Dateisystemdatenverwaltungsanordnung Speicherraumzuteilungsinformationen für die jeweilige Datei aufweist.

[0059] Es wird angemerkt, daß sowohl das Dateistatusattribut als auch das Referenzattribut der logischen Einheit extern definierte Attribute darstellen, da sie keine Attribute sind, die normal in einem Dateisystem definiert sind. So muß das Dateisystem extern definierte Dateiattribute unterstützen, um diese bevorzugte Form der Erfindung zu implementieren. Die meisten modernen Dateisysteme unterstützen derartige extern definierte Dateiattribute.

[0060] Bezug nehmend auf **Fig. 7** umfaßt die Datenverwaltungsanordnung für Arraysteuerungsprozesse **300** einen Antrag **701** für jede Datei, der gemäß der Erfindung direkt durch die Arraysteuerungsprozesse gehandhabt wird. Anders ausgedrückt wird ein Eintrag **701** für jede Datei, die einer umleitbaren Operation unterzogen wird, erzeugt. Jeder Eintrag **701** umfaßt ein Referenzfeld **702** der logischen Einheit und ein Dateinamenfeld **703**. Das Referenzfeld **702** der logischen Einheit speichert eine Referenz auf eine logische Einheit, bei der die bestimmte Datei gespeichert ist. Das Dateinamenfeld **703** speichert den Dateinamen, der der Datei zugeordnet ist, für die der Eintrag **701** gemacht wird. Der Eintrag **701** umfaßt ferner ein Feld **705**, das Speicherraumzuteilungsinformationen enthält, die die Blöcke von Speicherraum anzeigen, der für die jeweilige Datei zugeteilt ist.

[0061] Eine bevorzugte Form der Erfindung umfaßt eine Anordnung zum Unterbringen einer Veränderung bei einem Dateistatus. Diese Form der Erfindung ermöglicht es, daß eine Datei von einem Großdateistatus zu einem Kleindateistatus oder umgekehrt bewegt wird oder migriert. Diese Migration zwischen einem kleinen und einem großen Dateistatus kann auf eine Anzahl von Weisen erzielt werden. Die oben beschriebenen Dateistatus können z. B. erweitert werden, um einen „wachsenden“ Dateistatus und einen „schrumpfenden“ Dateistatus zu umfassen. Ein wachsender Dateistatus bedeutet, daß die Datei wie gemäß dem System definiert von einer kleinen Datei zu einer großen Datei wächst. Ein schrumpfender Dateistatus bedeutet, daß die Subjektdateteil einmal als eine große Datei behandelt wurde, nun jedoch ausreichend kleiner wird, um dieselbe zur Handhabung durch das Dateisystem als eine kleine Datei zu qualifizieren. Diese Migration kann auftreten, wenn die Datei modifiziert wird, oder als ein separater Prozeß, nachdem die Datei modifiziert wurde. Der Effekt der Veränderung des Status soll verändern, wie Dateisystemzugriffsoperationen, die die Datei spezifizieren, in der Zukunft gehandhabt werden, und welcher Pfad **401** oder **403** für Dateidaten bei zukünftigen Zugriffsoperationen auf die Datei genommen wird.

[0062] Die oben beschriebenen bevorzugten Ausführungsbeispiele sollen die Prinzipien der Erfindung

darstellen, den Bereich der Erfindung jedoch nicht einschränken. Verschiedene weitere Ausführungsbeispiele und Modifizierungen an diesen bevorzugten Ausführungsbeispielen können von Fachleuten durchgeführt werden, ohne von dem Bereich der folgenden Ansprüche abzuweichen. Insbesondere sollte es ersichtlich sein, daß die Anordnung von Verarbeitungsgeräten zum Ausführen der verschiedenen Programmprozesse gemäß der Erfindung nur zu Beispielzwecken gezeigt ist. Die verschiedenen Prozesse können auf jede geeignete Weise über Verarbeitungsgeräte verteilt sein, um die erwünschten Funktionen in dem Bereich der Erfindung und der folgenden Ansprüche durchzuführen. Außerdem sind die verschiedenen Softwarekomponenten, die in den Figuren gezeigt sind, nur zu Beispielzwecken gezeigt und beschrieben und sind allgemein auf eine Weise unterteilt, die die Erfindung sehr klar beschreibt. Die verschiedenen Softwarefunktionen können jedoch durch unterschiedlich benannte und angeordnete Softwarekomponenten durchgeführt werden. Diese unterschiedlichen Softwareanordnungen sollen als gleichwertig zu der Anordnung, die in der vorliegenden Offenbarung und den beigefügten Ansprüchen dargestellt ist, betrachtet werden.

Patentansprüche

1. Verfahren zum Zugreifen auf ein Datenspeichergerät (**203**), das durch Arraysteuerungsprozesse (**300**) gesteuert wird, wobei das Verfahren folgende Schritte umfaßt:
 - (a) Empfangen einer Dateisystemzugriffsoperationsanforderung;
 - (b) Bestimmen, ob die Dateisystemzugriffsoperationsanforderung eine Operation spezifiziert, die eine umleitbare Operation aufweist;
 - (c) falls die Dateisystemzugriffsoperationsanforderung eine umleitbare Operation spezifiziert, Umleiten der umleitbaren Operation von Dateisystemprozessen (**308**), die dem Datenspeichergerät (**203**) zugeordnet sind, zu den Arraysteuerungsprozessen (**300**);
 - (d) Durchführen der umleitbaren Operation mit den Arraysteuerungsprozessen (**300**); und
 - (e) Aktualisieren einer Dateisystemdatenverwaltungsanordnung mit Operationsergebnisinformationen von den Arraysteuerungsprozessen, wobei die Dateisystemdatenverwaltungsanordnung durch die Dateisystemprozesse (**308**) gesteuert wird.
2. Verfahren gemäß Anspruch 1, bei dem die Operation eine Erzeugen-Operation aufweist, und bei dem die Operationsergebnisinformationen eine Referenz auf den Arrayspeicherraum umfassen, an den Daten für die Datei, die in der Erzeugen-Operation spezifiziert ist, geschrieben werden sollen.
3. Verfahren gemäß Anspruch 2, bei dem der Schritt des Aktualisierens der Dateisystemdatenverwaltungsanordnung ein Bewirken dessen umfaßt, daß die Dateisystemprozesse (**308**) einen Eintrag (**601**) in die Dateisystemdatenverwaltungsanordnung erzeugen, was ein Identifizieren von Informationen für die spezifizierten Daten und eine Referenz auf den Arrayspeicherraum, an den die spezifizierten Daten geschrieben werden sollen, umfaßt.
4. Verfahren gemäß einem der Ansprüche 1 bis 3, bei dem die umleitbare Operation eine Lesen-Operation aufweist und ferner den Schritt des Lesens an einer Arrayspeicherraumreferenz für die Datei, die in der Lesen-Operation spezifiziert ist, umfaßt.
5. Verfahren gemäß Anspruch 4, bei dem:
 - (a) das Durchführen der umleitbaren Operation ein Lesen von Daten aus Blöcken, die dem Raum zugeteilt sind, der durch die Arrayspeichersystemraumreferenz identifiziert ist, aufweist; und
 - (b) die Operationsergebnisinformationen eine Anzeige umfassen, daß die Lesen-Operation abgeschlossen ist.
6. Verfahren gemäß Anspruch 5, bei dem der Schritt des Aktualisierens der Dateisystemdatenverwaltungsanordnung ein Bewirken dessen umfaßt, daß die Dateisystemprozesse (**308**) Zugriffsdaten in einem Antrag für die Datei, die in der Lesen-Operation spezifiziert ist, modifizieren.
7. Verfahren gemäß einem der Ansprüche 1 bis 6, bei dem der Schritt des Bestimmens, ob die Dateisystemzugriffsoperationsanforderung eine umleitbare Operation umfaßt, mit einem Keildateisystem (**310**) durchgeführt wird, das zwischen einem Dateisystemklienten (**212**) und den Dateisystemprozessen (**308**), die der Datenspeichervorrichtung zugeordnet sind, eingefügt ist.
8. Verfahren gemäß Anspruch 7, bei dem der Schritt des Bestimmens, ob die Dateisystemzugriffsoperationsanforderung eine umleitbare Operation umfaßt, ein Lesen eines Dateiattributs für die Datei, die in der Dateisystemzugriffsoperationsanforderung spezifiziert ist, aufweist.
9. Verfahren gemäß Anspruch 7, bei dem der Schritt des Bestimmens, ob die Dateisystemzugriffsoperationsanforderung eine umleitbare Operation umfaßt, ein Lesen einer Dateigröße, die in der Dateisystemzugriffsoperationsanforderung enthalten ist, aufweist.
10. Verfahren gemäß einem der Ansprüche 1 bis 9, das ferner den Schritt des Modifizierens der umleitbaren Operation in eine Form, die geeignet ist, um durch die Arraysteuerungsprozesse (**300**) durchgeführt zu werden, umfaßt.
11. Verfahren gemäß Anspruch 10, bei dem der

Schritt des Modifizierens der umleitbaren Operation ein Umwandeln der Operation von einer Byteversatzoperation in eine Blockzugriffsoperation aufweist.

12. Verfahren gemäß einem der Ansprüche 1 bis 11, bei dem der Schritt des Bestimmens, ob die Dateisystemzugriffsoperationsanforderung eine umleitbare Operation umfaßt, durch die Dateisystemprozesse (**308**) durchgeführt wird.

13. Verfahren gemäß einem der Ansprüche 1 bis 12, bei dem die umleitbare Operation eine Schreiben- oder eine Erzeugen-Operation aufweist und der Schritt des Durchführens der umleitbaren Operation mit den Arraysteuerungsprozessen (**300**) ein Weiterleiten von Daten für eine Datei, die in der umleitbaren Operation spezifiziert ist, direkt von einem Dateisystemklienten (**212**) zu den Arraysteuerungsprozessen (**300**) umfaßt.

14. Verfahren gemäß einem der Ansprüche 1 bis 12, bei dem die umleitbare Operation eine Lesen-Operation aufweist und der Schritt des Durchführens der umleitbaren Operation mit den Arraysteuerungsprozessen (**300**) ein Weiterleiten von Daten für eine Datei, die in der umleitbaren Operation spezifiziert ist, direkt von den Arraysteuerungsprozessen zu einem Dateisystemklienten (**212**) umfaßt.

15. Computerprogramm gemäß Anspruch 15, bei dem der Operationserfassungsprogrammcode einen Keildateisystemprogrammcode aufweist, der zwischen einem Dateisystemklienten (**212**) und dem Dateisystemprogrammcode betreibbar ist.

16. Computerprogramm gemäß Anspruch 15 oder 16, bei dem der Arraysteuerungsprogrammcode einen Bytezugriffsumwandlungsprogrammcode zum Umwandeln von Byteversatzzugriffsinstruktionen in einer umleitbaren Operation zu Blockzugriffsinstruktionen umfaßt.

17. Computerprogramm gemäß einem der Ansprüche 15 bis 17, bei dem der Operationserfassungsprogrammcode einen temporären Datenstrukturprogrammcode zum Erzeugen einer temporären Datenstruktur, die Informationen von der Dateisystemdatenverwaltungsanordnung enthält, auf einer Datei, die in einer empfangenen Zugriffsoperationsanforderung spezifiziert ist, umfaßt.

18. Computerprogramm gemäß einem der Ansprüche 15 bis 18, bei dem der Operationserfassungsprogrammcode in den Dateisystemprogrammcode eingebaut ist.

19. Computerprogramm gemäß einem der Ansprüche 15 bis 19, bei dem der Arraysteuerungsprogrammcode einen Dateisystemformatierungsprogrammcode zum Formatieren von Datenmeldungen

gemäß einem Dateisystemstandard für ein Dateisystem, das durch den Dateisystemprogrammcode implementiert ist, umfaßt.

20. Computerprogramm gemäß Anspruch 21, bei dem der Operationserfassungsprogrammcode einen Keildateisystemprogrammcode aufweist, der zwischen einem Dateisystemklienten (**212**) und dem Dateisystemprogrammcode betreibbar ist.

21. Computerprogramm gemäß Anspruch 21 oder 22, bei dem der Arraysteuerungsprogrammcode einen Bytezugriffsumwandlungsprogrammcode zum Umwandeln von Byteversatzzugriffsinstruktionen in einer umleitbaren Operation zu Blockzugriffsinstruktionen umfaßt.

22. Computerprogramm gemäß einem der Ansprüche 21 bis 23, bei dem der Operationserfassungsprogrammcode einen temporären Datenstrukturprogrammcode zum Erzeugen einer temporären Datenstruktur, die Informationen aus der Dateisystemverwaltungsanordnung enthält, auf einer Datei, die in einer empfangenen Zugriffsoperationsanforderung spezifiziert ist, umfaßt.

23. Computerprogramm gemäß einem der Ansprüche 21 bis 24, bei dem der Arraysteuerungsprogrammcode einen Dateisystemformatierungsprogrammcode zum Formatieren von Datenmeldungen gemäß einem Dateisystemstandard für ein Dateisystem, das durch den Dateisystemprogrammcode implementiert ist, umfaßt.

24. Datenspeichersystem (**201**) mit folgenden Merkmalen:

- (a) einem Datenspeichergerät (**203**);
- (b) einer Datenverarbeitungseinrichtung zum
 - (I) Bestimmen, ob eine Dateisystemzugriffsoperationsanforderung eine umleitbare Operation umfaßt,
 - (II) wenn die Dateisystemzugriffsoperationsanforderung eine umleitbare Operation umfaßt, Weiterleiten der umleitbaren Operation zu Arraysteuerungsprozessen, die durch die Datenverarbeitungseinrichtung ausgeführt werden,
 - (III) Durchführen der umleitbaren Operation mit den Arraysteuerungsprozessen (**300**), und
 - (IV) Aktualisieren einer Dateisystemdatenverwaltungsanordnung mit Operationsergebnisinformationen für die umleitbare Operation von den Arraysteuerungsprozessen;
- (c) einer Eingangs/Ausgangsordnung, die mit der Datenverarbeitungseinrichtung verbunden ist, zum Weiterleiten von Kommunikationen zwischen einem Dateisystemklienten (**212**) und der Datenverarbeitungseinrichtung; und
- (d) einer Schnittstelle zwischen der Datenverarbeitungseinrichtung und dem Datenspeichergerät (**203**).

25. Dateispeichersystem gemäß Anspruch 26,

bei der die Datenverarbeitungseinrichtung folgende Merkmale aufweist:

- (a) eine Arraysteuerung (**204**), die die Arraysteuerungsprozesse (**300**) ausführt; und
- (b) einen Dateisystemprozessor (**205**), der Dateisystemprozesse (**308**) ausführt und die Dateisystemdatenverwaltungsanordnung beibehält.

26. Dateispeichersystem gemäß Anspruch 26 oder 27, bei der die Eingangs/Ausgangsordnung folgende Merkmale umfaßt:

- (a) ein Eingangs/Ausgangs-Tor (**208**), das eine physische Eingangs/Ausgangsverbindung aufweist; und
- (b) eine Netzprotokollstapelkomponente (**210**), die betreibbar mit dem Eingangs/Ausgangs-Tor verbunden ist.

27. Dateispeichersystem gemäß Anspruch 28, bei der die Netzprotokollstapelkomponente (**210**) Netzschnittstelleneinrichtungen für sowohl die Arraysteuerung (**204**) als auch den Dateisystemprozessor (**205**) liefert.

28. Computerprogramm zum Bedienen von Dateisystemzugriffsoperationsanforderungen von einem Dateisystemklienten (**212**) entsprechend dem Verfahren nach Anspruch 1, wobei das Computerprogramm auf einem computerlesbaren Medium gespeichert ist und folgende Merkmale umfaßt:

- (a) einen Operationserfassungsprogrammcode zum Erfassen umleitbarer Operationen und nicht umleitbarer Operationen in einer empfangenen Dateisystemzugriffsoperationsanforderung;
- (b) einen Dateisystemprogrammcode zum Durchführen jeder nicht umleitbaren Operation, die in einer empfangenen Zugriffsoperationsanforderung enthalten ist, und zum Beibehalten einer Dateisystemdatenverwaltungsanordnung, die Informationen für jede Datei in dem Dateisystem umfaßt; und
- (c) einen Arraysteuerungsprogrammcode zum Durchführen jeder umleitbaren Operation, zum Verwalten eines Datenspeichergeräts (**203**), das ein Datenspeichermedium umfaßt, und zum Kommunizieren mit dem Dateisystemprogrammcode, um die Dateisystemdatenverwaltungsanordnung entsprechend auf das Verhalten einer umleitbaren Operation zu aktualisieren.

29. Computerprogramm zum Bedienen von Dateisystemzugriffsoperationsanforderungen von einem Dateisystemklienten (**212**) zu einem Dateisystem, das eine Dateisystemdatenverwaltungsanordnung umfaßt, die Informationen auf jeder Datei in dem Dateisystem enthält, entsprechend dem Verfahren nach Anspruch 1, wobei das Computerprogramm auf einem computerlesbaren Medium gespeichert ist und folgende Merkmale umfaßt:

- (a) einen Operationserfassungsprogrammcode zum Erfassen umleitbarer Operationen in einer Dateisystemzugriffsoperationsanforderung, die von

einem Dateisystemklienten (**212**) empfangen wird, und zum Verhindern, daß jede umleitbare Operation durch das System ausgeführt wird; und

(b) einen Arraysteuerungsprogrammcode zum Durchführen jeder umleitbaren Operation, zum Verwalten eines Datenspeichergeräts (**203**), das ein Datenspeichermedium umfaßt, und zum Kommunizieren mit dem Dateisystemprogrammcode, um die Dateisystemdatenverwaltungsanordnung entsprechend auf das Verhalten einer umleitbaren Operation zu aktualisieren.

Es folgen 6 Blatt Zeichnungen

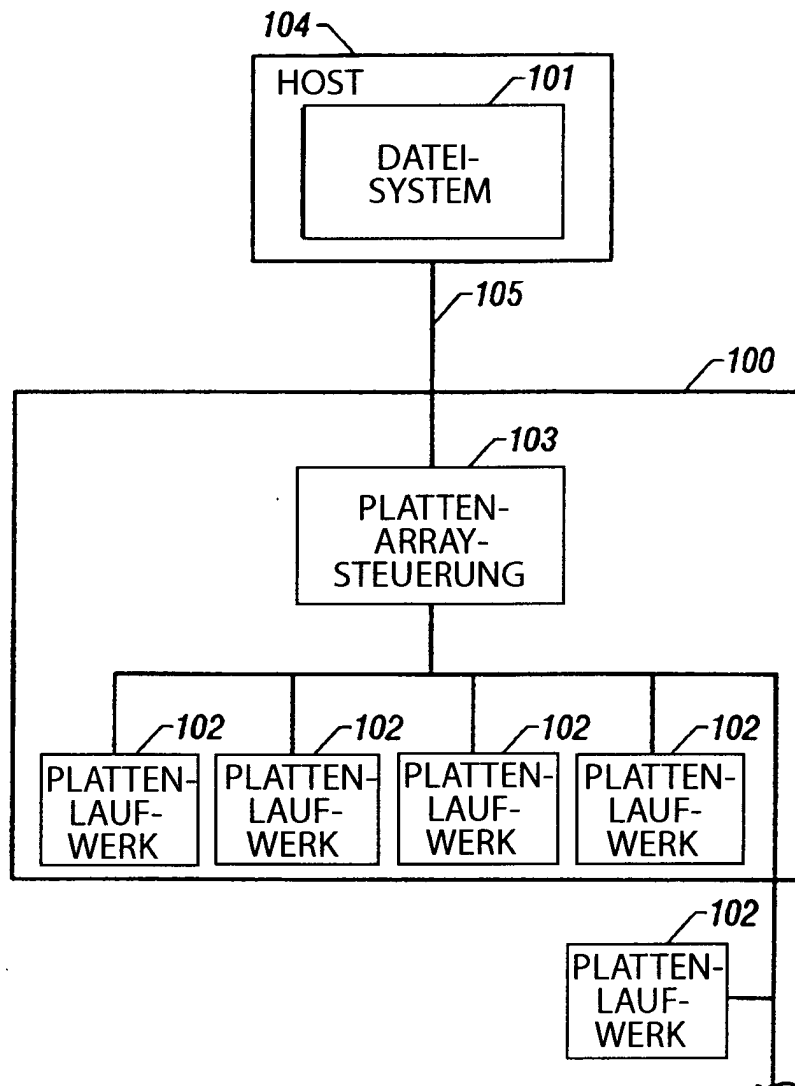


FIG 1
(STAND DER TECHNIK)

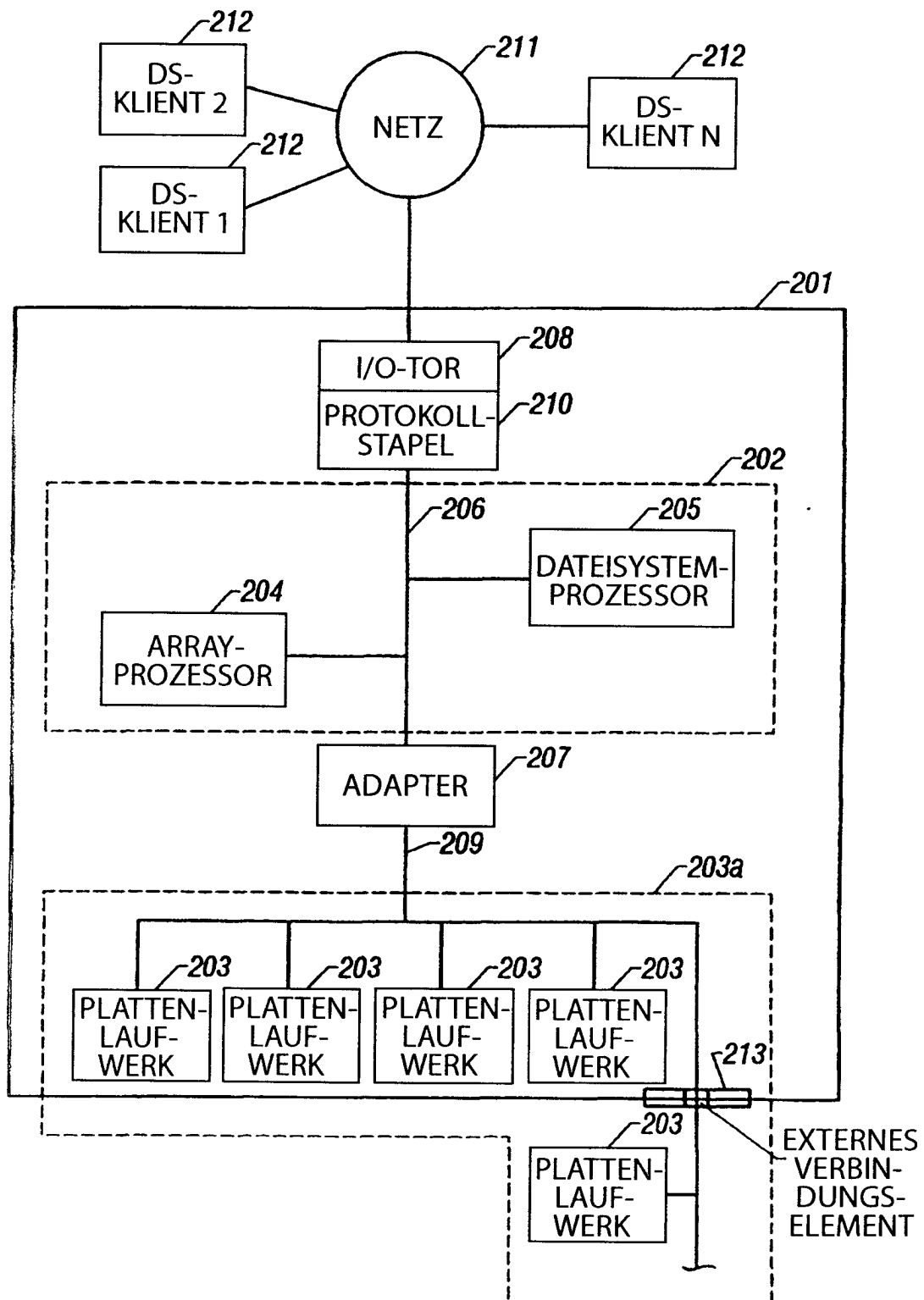


FIG 2

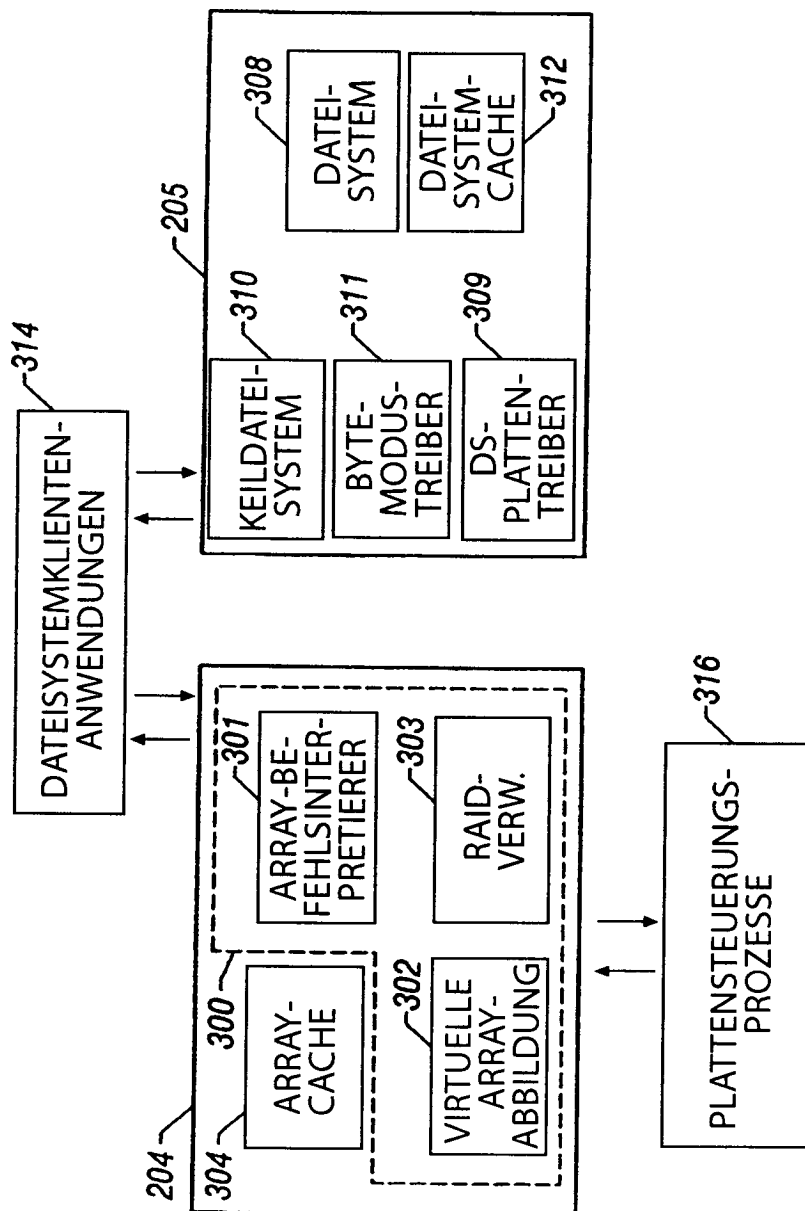


FIG 3

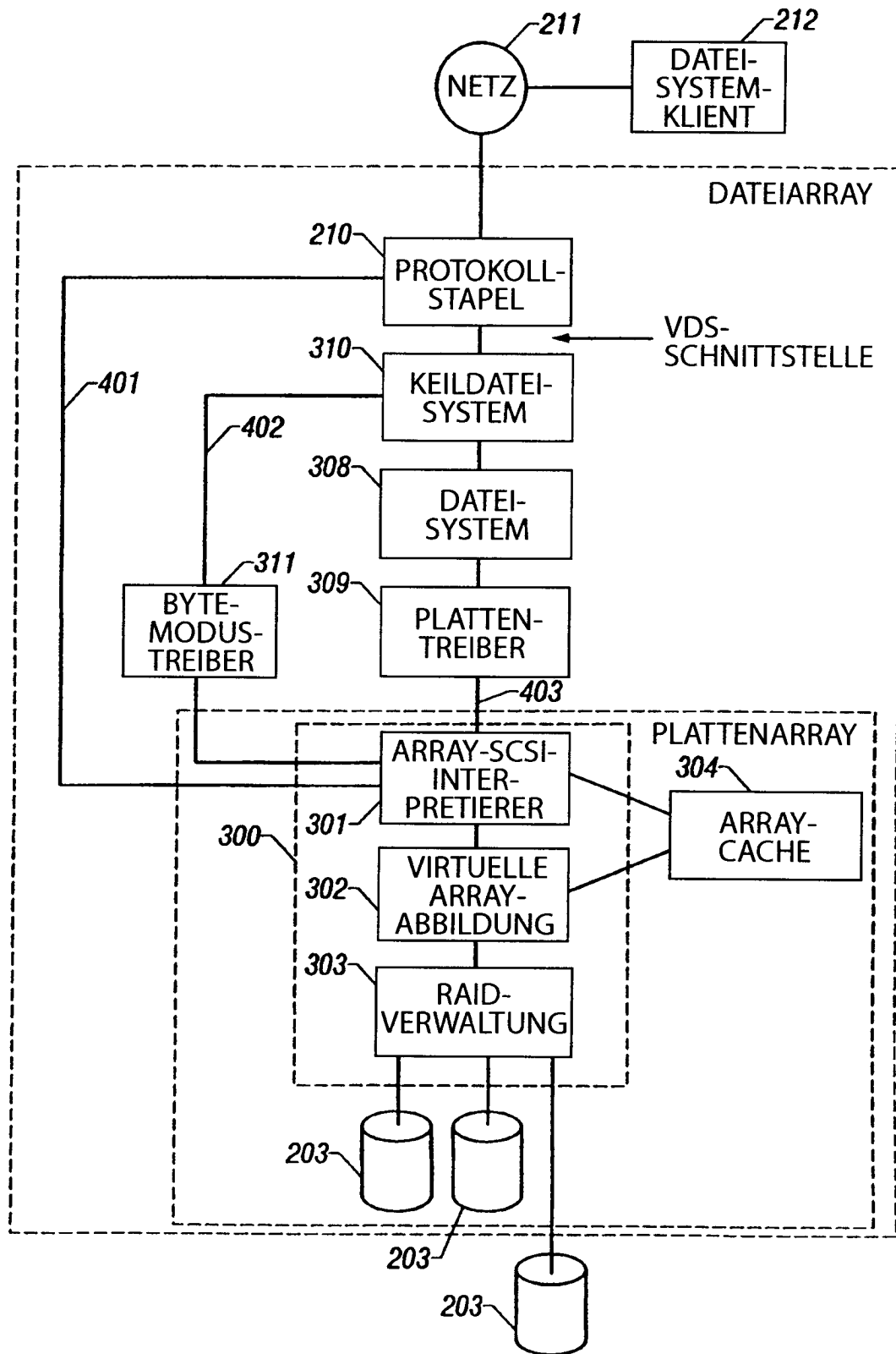


FIG 4

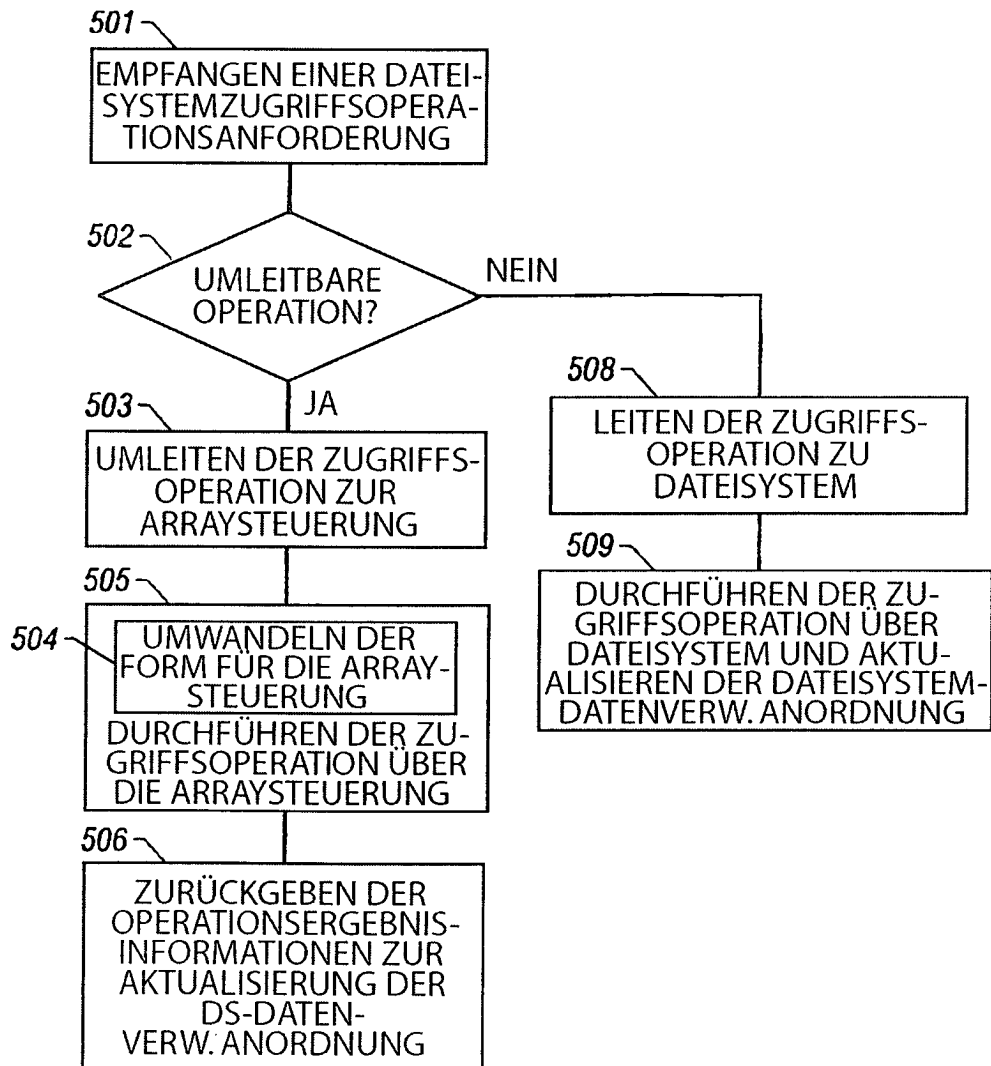


FIG 5

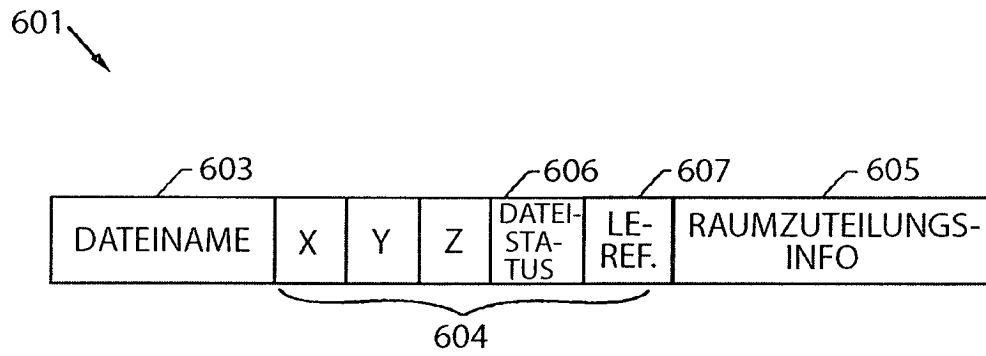


FIG 6

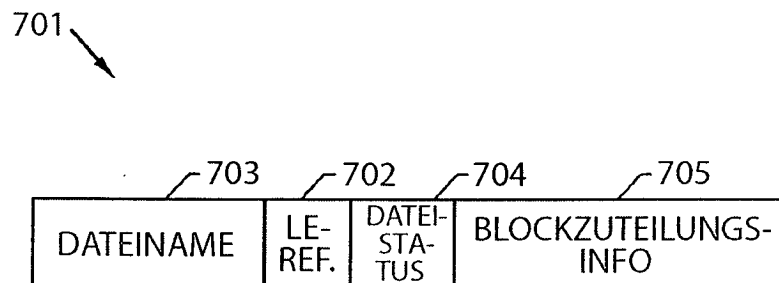


FIG 7