(19) Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 101 24 482 B4** 2005.01.27

(12) **Patentschrift**

(21) Aktenzeichen: **101 24 482.7**
(22) Anmeldetag: **19.05.2001**
(43) Offenlegungstag: **06.12.2001**
(45) Veröffentlichungstag der Patenterteilung: **27.01.2005**

(51) Int Cl.[7]: **G06F 12/16**

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

(30) Unionspriorität:
580539 26.05.2000 US

(71) Patentinhaber:
EMC Corporation, Hopkinton, Mass., US

(74) Vertreter:
Lorenz und Kollegen, 89522 Heidenheim

(72) Erfinder:
MacHardy jun., Earle Trounson, Durham, N.C., US;
Forest, Miles Aram de, Bahama, N.C., US

(56) Für die Beurteilung der Patentfähigkeit in Betracht gezogene Druckschriften:
US 59 87 621

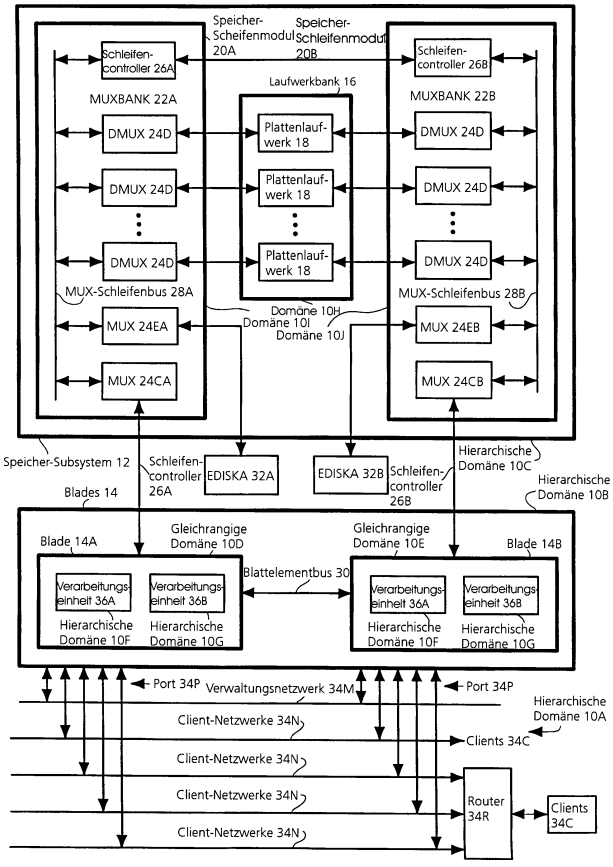
(54) Bezeichnung: **Fehlertolerante Systemressource mit niedriger Latenzzeit, mit übergeordneter Protokollierung von Systemressourcentransaktionen und serverübergreifend gespiegelter Protokollierung von übergeordneten Systemressourcentransaktionen**

(57) Hauptanspruch: Übergeordneter Transaktionsprotokollierungsmechanismus zur Verwendung in Verbindung mit einer gemeinsam genutzten Systemressource (10) mit einem Ressourcen-Subsystem (12) zur Ausführung von untergeordneten Systemressourcenvorgängen und einem Steuerungs-/Verarbeitungssubsystem (14) mit einer ersten und einer zweiten Sub-Rechnereinheit (14A,14B), die jeweils einen Systemressourcenprozessor (80A,80B) aufweisen, der übergeordnete Systemressourcenvorgänge ausführt und Systemressourcenanforderungen von Clients (34C,74C) in entsprechende untergeordnete Systemressourcenvorgänge umwandelt, der in der ersten und in der zweiten Sub-Rechnereinheit (14A,14B) des Steuerungs-/Verarbeitungssubsystems (14) jeweils folgendes aufweist:

– einen in der Sub-Rechnereinheit (14A,14B) angeordneten Protokollerzeuger (50G,92LG) zur Gewinnung von übergeordneten Vorgangsinformationen in Bezug auf jeden übergeordneten Vorgang der Sub-Rechnereinheit (14A,14B); und

– ein in der Sub-Rechnereinheit (14A,14B) angeordnetes Transaktionsprotokoll zur Speicherung der übergeordneten Vorgangsinformationen, wobei der protokollierende Mechanismus zur Wiederherstellung des Betriebes der Sub-Rechnereinheit (14A,14B) nach einem Ausfall der Sub-Rechnereinheit (14A,14B) für das Lesen der übergeordneten Vorgangsinformationen aus dem Transaktionsprotokoll und das Wiederherstellen des Ausführungsstatus der Sub-Rechnereinheit (14A,14B) verantwortlich ist; und

– ein Transaktionsprotokoll-Spiegelungsmechanismus, der...



Beschreibung

[0001] Die Erfindung betrifft einen übergeordneten Transaktionsprotokollierungsmechanismus zur Verwendung in Verbindung mit einer gemeinsam genutzten Systemressource und ein Verfahren zur Protokollierung von Systemressourcentransaktionen und zur Wiederherstellung des Ausführungsstatus von Systemressourcenanforderungen in einer Clients Systemressourcendienste bietenden Systemressource.

Stand der Technik

[0002] Ein ständiges Problem bei Computersystemen besteht in der Bereitstellung sicherer, fehlertoleranter Ressourcen, wie beispielsweise Kommunikations- und Datenspeicherressourcen, so dass die Kommunikationen zwischen dem Computersystem und den Clients oder Benutzern des Computersystems bei einem Ausfall aufrechterhalten werden und das so beschaffen ist, dass Daten nicht verlorengehen, und bei einem Ausfall ohne Verlust wiederhergestellt oder rekonstruiert werden können. Dieses Problem ist insbesondere bei Netzwerksystemen gravierend, bei denen eine gemeinsam genutzte Ressource, wie beispielsweise eine Systemdatenspeichereinrichtung, typischerweise aus einer oder mehreren Systemressourcen, wie beispielsweise Dateiservern besteht, die von einer Anzahl von Clients gemeinsam genutzt, und auf die über das Systemnetzwerk zugegriffen wird. Ein Ausfall bei einer gemeinsam genutzten Ressource, wie beispielsweise bei den Datenspeicherfunktionen eines Dateiservers oder bei den Kommunikationen zwischen Clients des Dateiservers und den von dem Dateiserver unterstützten Client-Dateisystemen, kann einen Ausfall des gesamten Systems zur Folge haben. Dieses Problem ist insbesondere dahingehend gravierend, dass der Umfang der Daten und der Kommunikationen und der Anzahl der durch eine gemeinsam genutzte Systemressource, wie beispielsweise einen Dateiserver, unterstützten Datentransaktionen bedeutend größer ist als bei einem System mit einem einzelnen Client, was eine bedeutend erhöhte Komplexität bei der Ressource, den Datentransaktionen und bei den Client-Serverkommunikationen zur Folge hat. Diese erhöhte Komplexität hat wiederum eine erhöhte Ausfallwahrscheinlichkeit und einen erhöhten Schwierigkeitsgrad bei der Wiederherstellung nach Ausfällen zur Folge. Zusätzlich ist das Problem dahingehend mehrdimensional, dass ein Ausfall bei jeder beliebigen einer Anzahl von Ressourcenkomponenten oder der damit in Zusammenhang stehenden Funktionen auftreten kann, wie beispielsweise bei einem Plattenlaufwerk, bei einem Steuerprozessor oder bei den Netzwerkkommunikationen. Außerdem ist es wünschenswert, dass die gemeinsam genutzten Ressourcenkommunikationen und -dienste trotz Ausfällen bei einer oder mehreren Komponenten weiterhin verfügbar bleiben, und dass die Vorgänge der Ressource, d.h. abgeschlossene Vorgänge und Transaktionen erhalten bleiben und wiederhergestellt werden, und auch Vorgänge und Transaktionen, die gerade ausgeführt werden, wenn ein Ausfall eintritt.

[0003] Unter Berücksichtigung von Netzwerk-Dateiserversystemen als ein typisches Beispiel einer gemeinsam genutzten Systemressource nach dem Stand der Technik ist bei den Dateiserversystemen nach den Stand der Technik eine Anzahl von Verfahren zum Erreichen von Fehlertoleranz bei Client-Serverkommunikationen und bei Dateitransaktionsfunktionen des Dateiservers, und bei der Datenwiederherstellung oder -rekonstruktion übernommen worden. Diese Verfahren basieren typischerweise auf Redundanz, d.h. auf der Bereitstellung duplizierter Systemelemente und dem Austausch eines ausgefallenen Elementes gegen ein dupliziertes Element oder der Erzeugung duplizierter Kopien von Informationen, die für die Rekonstruktion verlorener Informationen verwendet werden sollen.

[0004] Beispielsweise ist bei vielen Systemen nach den Stand der Technik eine RAID-Technologie nach Industriestandard zur Aufrechterhaltung und Wiederherstellung von Daten- und Dateitransaktionen integriert, wobei die RAID-Technologie eine Verfahrensfamilie zur Verteilung redundanter Daten- und Fehlerkorrekturinformationen über eine redundante Anordung von Plattenlaufwerken ist. Ein ausgefallenes Plattenlaufwerk ist durch ein redundantes Laufwerk ersetzbar, und die Daten auf dem ausgefallenen Laufwerk können aus den redundanten Daten- und Fehlerkorrekturinformationen rekonstruiert werden. Bei anderen Systemen nach dem Stand der Technik werden mehrfache, duplizierte parallele Kommunikationspfade oder mehrfache, duplizierte parallele Verarbeitungseinheiten mit passenden Schaltungen verwendet, um Kommunikationen oder Dateitransaktionen von einem ausgefallenen Kommunikationspfad oder Dateiprozessor zu einem gleichwertigen parallelen Pfad oder Prozessor zu schalten, um die Zuverlässigkeit und Verfügbarkeit von Client-Serverkommunikationen und Client-Client-Dateisystemkommunikationen zu erweitern. Diese Verfahren sind jedoch bei Systemressourcen kostspielig, weil die Duplizierung wesentlicher Kommunikationspfade und Verarbeitungspfade, und die Einbindung komplexer Verwaltungs- und Synchronisationsmechanismen zur Verwaltung des Austausches ausgefallener Elemente durch funktionierende Elemente erforderlich ist. Außerdem, und während diese Verfahren eine Fortsetzung der Dienste und Funktionen bei einem Ausfall ermöglichen, und RAID-Verfahren beispielsweise die Wiederherstellung oder Rekonstruktion abgeschlossener Datentransaktionen ermöglichen, d.h. Transaktionen; die zur festen Speicherung auf der Platte festgeschrieben wurden, unterstützen diese Verfahren nicht die Rekonstruktion oder Wiederherstellung von Transaktionen, die aufgrund von Ausfäl-

len während der Ausführung der Transaktionen verlorengingen.

[0005] Als eine Folge wird auch bei anderen Verfahren nach dem Stand der Technik Informationsredundanz verwendet, um die Wiederherstellung und Rekonstruktion von Transaktionen zu ermöglichen, die aufgrund von während der Ausführung der Transaktionen eintretenden Ausfällen verlorengingen. Diese Verfahren umfassen die Pufferung, Transaktionsprotokollierung und Spiegelung, wobei es sich bei der Pufferung um die temporäre Speicherung von Daten im Speicher in dem Datenflusspfad zu, und von dem Festspeicher handelt, bis die Datentransaktion durch Übertragung der Daten zur Speicherung im Festspeicher festgeschrieben ist, d.h. in einem Plattenlaufwerk, oder von dem Festspeicher gelesen, und an einen Empfänger übertragen wird. Bei der Transaktionsprotokollierung oder -aufzeichnung werden eine Datentransaktion beschreibende Informationen temporär gespeichert, d.h. der angeforderte Dateiservervorgang, bis die Datentransaktion zur Speicherung im Festspeicher festgeschrieben ist, d.h. in dem Dateiserver abgeschlossen ist, wodurch eine Rekonstruktion oder erneute Ausführung verlorener Datentransaktionen aus den gespeicherten Informationen ermöglicht wird. Die Spiegelung wird dagegen oftmals in Verbindung mit der Pufferung oder der Transaktionsprotokollierung verwendet, wobei es sich im wesentlichen um die Speicherung einer Kopie der Inhalte einer Pufferspeicher- oder Transaktionsprotokollanmeldung handelt, wie beispielsweise der Speicher oder feste Speicherplatz eines separaten Prozessors, wenn die Pufferspeicher- oder Transaktionsprotokolieinträge in dem Dateiprozessor erzeugt werden.

[0006] Die Pufferung, Transaktionsprotokollierung und Spiegelung sind oftmals nicht zufriedenstellend, da sie bei Systemressourcen kostspielig sind und komplexe Verwaltungs- und Synchronisationsvorgänge und -mechanismen zur Verwaltung der Pufferungs-, Transaktionsprotokollierungs- und Spiegelungsfunktionen und der nachfolgenden Transaktionswiederherstellungsvorgänge erfordern und die Dateiserverlatenzzeit bedeutend erhöhen, d.h. die Zeit, die zum Abschluss einer Dateitransaktion erforderlich ist. Es muss auch bemerkt werden, dass Pufferung und Transaktionsprotokollierung anfällig für Ausfälle bei den Prozessoren ist, in denen die Pufferungs- und Protokollierungsmechanismen resident sind, und dass, während die Spiegelung eine Lösung für das Problem des Verlustes der Pufferungs- oder Transaktionsprotokollierungsinhalte ist, die Spiegelung ansonsten unter denselben Nachteilen leidet, wie die Pufferung oder Transaktionsprotokollierung. Diese Probleme stehen dahingehend miteinander in Verbindung, dass die Pufferung, und insbesondere die Transaktionsprotokollierung und Spiegelung die Speicherung bedeutender Informationsmengen erfordern, während die Transaktionsprotokollierung und die Rekonstruktion oder erneute Ausführung protokollierter Dateitransaktionen die Realisierung und Ausführung komplexer Algorithmen zur Analysierung, Wiederholung und Zurücklaufen lassen des Transaktionsprotokolls zur Rekonstruktion der Dateitransaktionen erfordern. Diese Probleme stehen weiterhin noch mehr miteinander dahingehend in Zusammenhang, dass diese Verfahren typischerweise in den niedrigeren Niveaus der Dateiserverfunktionen angesiedelt sind, wo jede Datentransaktion als eine große Anzahl detaillierter komplexer Dateisystemvorgänge ausgeführt wird. Als Folge wird die zu gewinnende und zu speichernde Informationsmenge und die Anzahl und die Komplexität der zur Gewinnung und Speicherung der Daten oder Datentransaktionen, und zur Wiederherstellung und Rekonstruktion der Daten oder Datentransaktionsvorgänge erforderlichen Vorgänge bedeutend erhöht.

[0007] Erneut muss hervorgehoben werden, dass diese Verfahren kostspielig bei Systemressourcen sind, und komplexe erwaltungs- und Synchronisationsmechanismen zur Verwaltung der Verfahren erfordern, und aufgrund der Kosten bei den Systemressourcen ist der Redundanzgrad, der durch diese Verfahren bereitgestellt werden kann, begrenzt, so dass die Systeme oftmals nicht vielfache Ausfallquellen bewältigen können. Ein System kann beispielsweise duplizierte parallele Prozessoreinheiten oder Kommunikationspfade für bestimmte Funktionen bereitstellen, aber das Eintreten von Ausfällen in beiden Prozessoreinheiten oder Kommunikationspfaden hat doch einen Totalverlust des Systems zur Folge. Zusätzlich arbeiten diese Verfahren zur Sicherstellung von Kommunikationen und der Aufrechterhaltung von Daten und Wiederherstellung typischerweise isoliert voneinander, und auf getrennten Niveaus oder Subsystemen. Aus diesem Grund arbeiten die Verfahren im allgemeinen nicht zusammen oder in Kombination miteinander, kollidieren möglicherweise beim Betrieb sogar miteinander, und können vielfache Ausfälle oder Kombinationen von Ausfällen oder Ausfälle, für deren Bewältigung eine Kombination von Verfahren notwendig ist, nicht bewältigen. Bei einigen Systemen nach dem Stand der Technik wird versucht, dieses Problem zu lösen, doch dies erfordert typischerweise die Verwendung eines zentralen Hauptkoordinationsmechanismus oder Subsystems und damit in Zusammenhang stehender Verwaltungs- und Synchronisationsmechanismen, um einen gemeinsamen Betrieb zu erreichen und Konflikte zwischen den Fehlerbehandlungsmechanismen zu vermeiden, was wiederum kostspielig bei den Systemressourcen, und in sich selbst eine Ausfallquelle ist.

[0008] Aus der US 5,987,621 ist ein Dateiserver mit zwei Streamserver-Computern, welche ein Disk-Array-Speichersubsystem mit einem Datennetzwerk verbinden, und mit wenigstens zwei Controllerser-

vern zum Empfang von Dateizugriffsanforderungen von Netzwerk-Clients, bekannt.

Aufgabenstellung

[0009] Der vorliegenden Erfindung liegt somit die Aufgabe zugrunde, einen übergeordneten Transaktionsprotokollierungsmechanismus zur Verwendung in Verbindung mit einer gemeinsam genutzten Systemressource und ein Verfahren zur Protokollierung von Systemressourcentransaktionen und zur Wiederherstellung des Ausführungsstatus von Systemressourcenanforderungen in einer Clients Systemressourcendienste bietenden Systemressource zu schaffen, die die Nachteile des Standes der Technik vermeiden, die insbesondere fehlertolerant und ressourcenschonend mit niedriger Latenzzeit auch häufige Client-Anfragen bearbeiten können und auch auf multible Ausfälle oder Kombinationen von Ausfällen Transaktionsausführungszustände wiederherstellen können.

[0010] Diese Aufgabe wird erfindungsgemäß durch die Ansprüche 1 und 4 gelöst.

Ausführungsbeispiel

[0011] Weitere Einzelheiten und Vorteile der vorliegenden Erfindung sind offensichtlich anhand der nachfolgenden Beschreibung der Erfindung und deren Ausführungen, wie in den dazugehörigen Figuren veranschaulicht, wobei:

[0012] **Fig.** 1 ein Blockdiagramm des Netzwerk-Dateiservers ist, in dem die vorliegende Erfindung realisiert werden kann;

[0013] **Fig.** 2 ein Blockdiagramm eines Prozessorkerns einer Domäne des Dateiservers von **Fig.** 1 ist;

[0014] **Fig.** 3 eine detailliertere schematische Darstellung einer Domäne des Dateiservers von **Fig.** 1 ist; und

[0015] **Fig.** 4 ein detailliertes Blockdiagramm der vorliegenden Erfindung ist.

Allgemeine Beschreibung der gemeinsam genutzten Hochverfügbarkeitsressource (**Fig.** 1):

[0016] Wie nachfolgend beschrieben wird, betrifft die vorliegende Erfindung eine Hochverfügbarkeitsressource, wie beispielsweise einen Dateiserver, Kommunikationsserver oder Druckserver, der von einer Anzahl von Benutzern in einem Netzwerksystem gemeinsam benutzt wird. Eine Ressource der vorliegenden Erfindung besteht aus einem integrierten zusammenwirkenden Cluster hierarchischer und gleichrangiger Domänen, wobei jede Domäne eine oder mehrere damit in Zusammenhang stehende, oder in den durch die Ressource unterstützten Funktionen oder Dienste integrierten Funktionen ausführt oder bereitstellt, wobei eine Domäne aus Subdomänen bestehen, oder diese umfassen kann. Beispielsweise können eine oder mehrere Domänen Kommunikationsdienste zwischen der Ressource und Netzwerk-Clients bereitstellen, andere Domänen können übergeordnete Dateisystem-, Kommunikations- oder Druckfunktionen ausführen, während andere Domänen untergeordnete Dateisystem-, Kommunikations- und Druckfunktionen ausführen können. Bei hierarchisch in Zusammenhang stehenden Domänen kann eine Domäne eine andere steuern oder kann eine übergeordnete oder untergeordnete Domäne durch Ausführung damit in Zusammenhang stehender übergeordneter oder untergeordneter Funktionen unterstützen. Beispielsweise kann eine übergeordnete Domäne übergeordnete Datei- oder Kommunikationsfunktionen ausführen, während eine damit in Zusammenhang stehende untergeordnete Domäne untergeordnete Datei- oder Kommunikationsfunktionen ausführen kann. Gleichrangige Domänen können im Gegensatz dazu identische oder parallele Funktionen ausführen, beispielsweise um die Kapazität der Ressource hinsichtlich bestimmter Funktionen durch Teilung der Aufgabenlast zu erhöhen, oder damit in Zusammenhang stehende Aufgaben oder Funktionen in gegenseitiger Unterstützung ausführen, um gemeinsam eine Domäne zu umfassen. Auch andere Domänen können hinsichtlich bestimmter Funktionen gleichrangige Domänen, und hierarchisch in Zusammenhang stehende Domänen hinsichtlich anderen Funktionen sein. Schließlich, und wie in den nachfolgenden Abhandlungen beschrieben wird, umfassen bestimmte Domänen Fehlerbehandlungsmechanismen, die separat und unabhängig von Fehlerbehandlungsmechanismen anderer Domänen funktionieren, jedoch zusammenwirkend, um ein hohes Niveau an Ressourcenverfügbarkeit zu erreichen.

[0017] Die vorliegende Erfindung kann beispielsweise und zum Zwecke der nachfolgenden Beschreibung in einen Hochverfügbarkeits-Netzwerkdateiserver (HAN File Server) **10** realisiert werden, wobei diese Realisierung in den nachfolgenden Abhandlungen als eine beispielhafte Ausführung der vorliegenden Erfindung detailliert beschrieben wird. Wie in **Fig.** 1 veranschaulicht, ist ein HAN File Server **10**, in dem die vorliegende Erfindung beispielsweise realisiert sein kann, beispielsweise ein Data General Corporation CLARiiON™-Dateiserver, der Clients durch die Verwendung eines Journaled File System Hochverfügbarkeits-Dateisystemanteile, d.h. Speicherplatz, Netzwerk-Failover-Fähigkeiten und eine Back-End-Redundant Array of Inexpensive Disks (RAID)-Speicherung von Daten bereitstellt. Bei einer gegenwärtig bevorzugten realisierten Ausführung unterstützt ein HAN File Server **10** sowohl das Common Internet File System Protokoll (CIFS), als auch gemeinsam genutzte Network File System (NFS)-Antei-

le nach Industriestandard, wobei die für die Dateizugriffssteuerung verwendeten gegensätzlichen Modelle, wie sie bei CIFS und NFS verwendet werden, transparent realisiert sind. Ein HAN File Server **10** ist auch in bestehende Verwaltungsdatenbanken nach Industriestandard integrierbar, wie beispielsweise Domänen-Controller in einer Microsoft Windows NT-Umgebung oder Network File System (NFS)-Domänen für Unix-Umgebungen.

[0018] Die gegenwärtig bevorzugte Realisierung bietet eine hohe Leistung durch Verwendung eines Nullkopie-IP-Protokollprofils, und zwar durch dichte Integration der Pufferungsverfahren des Dateisystems mit den Back-End-RAID-Mechanismen sowie durch die Verwendung eines Doppelspeicherprozessors, um eine Verfügbarkeit kritischer Daten durch Spiegelung auf dem gleichrangigen Speicherprozessor bereitzustellen und die Notwendigkeit von Schreibvorgängen auf eine Speicherplatte zu vermeiden. Wie nachfolgend im Detail beschrieben werden wird, arbeitet ein HAN File Server **10** der gegenwärtig bevorzugten Realisierung in einem Doppelprozessor bzw. in einem funktionellen Mehrfachverarbeitungsmodus, bei dem ein Prozessor als ein Datenstationsrechner arbeitet, um alle Netzwerk- und Dateisystemvorgänge zur Übertragung von Daten zwischen den Clients und dem plattenspeicherresidenten Dateisystem auszuführen und unterstützt einen Netzwerkstapel, eine CIFS/NFS-Realisierung und ein Journaled File System. Der zweite Prozessor fungiert als ein Blockspeicherprozessor, um alle Aspekte des Schreibens und Lesens von Daten in und aus einer Sammlung von in einer Hochverfügbarkeits-RAID-Konfiguration verwalteten Platten auszuführen.

[0019] Bei der gegenwärtig bevorzugten Realisierung ist das Dateisystem als ein aufzeichnendes Sofortwiederherstellungssystem mit einem auf einem Kernel basierenden CIFS-Netzwerkstapel realisiert und unterstützt NFS-Vorgänge in einem zweiten Modus, ist jedoch gemäß der vorliegenden Erfindung abgeändert, um einen Hochverfügbarkeitszugriff auf die Daten im Dateisystem bereitzustellen. Das Dateisystem bietet weiterhin Schutz vor Verlust eines Speicherprozessors durch den Erhalt aller Datenänderungen, die Netzwerk-Clients an dem Dateisystem mittels eines Datenreflexionsmerkmales vornehmen, bei dem alle im Speicher eines Speicherprozessors gespeicherten Datenänderungen bei Hardware- oder Softwareausfällen des Speicherprozessors erhalten bleiben. Die Reflexion von kerninternen Datenänderungen im Dateisystem wird durch ein Zwischenspeicher-Prozessorkommunikationssystem erzielt, wobei Datenänderungen in dem Dateisystem durch Clients auf einem Speicherprozessor mitgeteilt werden, wobei entweder das NFS oder CIFS verwendet, und durch den anderen Speicherprozessor als empfangen reflektiert und anerkannt wird, bevor eine positive Rückmeldung an den die Daten speichernden Netzwerk-Client geschickt wird. Dies stellt sicher, daß eine Kopie der Datenänderung auf dem alternativen Speicherprozessor aufgenommen wird, wenn beim Original-Speicherprozessor ein Ausfall eintritt, und sofern dieser Ausfall eintritt, die Änderungen auf das Dateisystem angewandt werden, nachdem dessen Betrieb durch das Failover-Verfahren auf den alternativen Speicherprozessor übertragen wurde. Wie beschrieben werden wird, ist dieser Reflexionsmechanismus oben auf darunterliegenden Dateisystem-Wiederherstellungsmechanismen aufgebaut, die so arbeiten, daß zur Rückverfolgung von Dateien verwendete System-Metadaten wiederhergestellt und repariert werden können, während der Reflexionsmechanismus Mechanismen zur Wiederherstellung oder zur Reparatur von Benutzerdaten bereitstellt. Das Blockspeichersubsystem bietet umgekehrt auf Plattenniveau Schutz vor dem Verlust einer Platteneinheit durch Verwendung der RAID-Technologie. Wenn ein Plattenlaufwerk verlorengegangen ist, bietet der RAID-Mechanismus den Mechanismus zum Wiederaufbau der Daten auf einem Ersatzlaufwerk, und bietet beim Betrieb Zugriff auf die Daten ohne das verlorene Plattenlaufwerk.

[0020] Wie beschrieben werden wird, stellt ein HAN File Server **10** der gegenwärtig bevorzugten Ausführung Hochverfügbarkeitskommunikationen zwischen Clients des Servers und den auf dem Server unterstützten Client-Dateisystemen durch redundante Komponenten und Datenpfade und Kommunikationsausfallbehandlungsmechanismen bereit, um die Kommunikationen zwischen Clients und Client-Dateisystemen aufrecht zu erhalten. Ein HAN File Server **10** der vorliegenden Erfindung weist ebenso Dateitransaktions-, Datensicherungs- und Wiederherstellungssysteme zur Verhinderung von Verlusten von Dateitransaktionen auf, und um die Wiederherstellung oder Rekonstruktion von Dateitransaktionen und Daten zu ermöglichen. Bei einem Systemhardware- oder Softwareausfall übernehmen die noch funktionsfähigen Komponenten des Systems die Aufgaben der ausgefallenen Komponente. Beispielsweise hat der Verlust eines einzelnen Ethernet-Ports auf einem Speicherprozessor zur Folge, daß der Netzwerkbetrieb von diesem Port durch einen anderen Port auf dem alternativen Speicherprozessor übernommen wird. Auf ähnliche Art und Weise würde der Verlust eines beliebigen Teils eines Speicherprozessors, der einen beliebigen Aspekt seiner Vorgänge beeinträchtigen würde, eine Übertragung des gesamten Netzwerkbetriebes und der Dateisysteme auf den noch funktionsfähigen Speicherprozessor zur Folge haben. Als weiteres Beispiel ist zu erwähnen, daß die Daten- und Dateitransaktions- und Sicherungsmechanismen die Wiederherstellung und Rekonstruktion von Daten- und Dateitransaktionen entweder durch die ausgefallene Komponente, wenn diese wiederhergestellt ist, oder durch eine entsprechende Komponente ermöglicht,

und es einer noch funktionierenden Komponente ermöglichen wird, die Dateitransaktionen einer ausgefallenen Komponente zu übernehmen. Zusätzlich hat der Verlust eines einzelnen Plattenlaufwerkes nicht den Verlust des Zugriffs auf die Daten zur Folge, weil die RAID-Mechanismen die noch funktionierenden Platten verwenden, um Zugriff auf die rekonstruierten Daten bereitzustellen, die sich auf dem verlorenen Laufwerk befanden. Bei Stromausfällen, die den gesamten Dateiserver betreffen, bleibt der Dateiserverstatus zum Zeitpunkt des Stromausfalls erhalten, und die im Speicher befindlichen Daten werden zur festen Speicherung festgeschrieben und wiederhergestellt, wenn die Stromzufuhr wieder einsetzt, wodurch alle vor dem Stromausfall vorgenommenen Datenänderungen erhalten bleiben. Schließlich sind die Kommunikations- und Daten- und Dateitransaktionsausfall-Wiederherstellungsmechanismen des HAN File Servers **10** in jeder Domäne oder Subsystem des Servers angeordnet und arbeiten separat und unabhängig voneinander, jedoch zusammenwirkend, um ein hohes Niveau an Verfügbarkeit von Client-Dateisystemkommunikationen zu erreichen und Verluste zu verhindern, und eine Wiederherstellung von Daten- und Dateitransaktionen zu ermöglichen. Die Ausfallwiederherstellungsmechanismen eines HAN File Servers **10** vermeiden jedoch die komplexen Mechanismen und Verfahren, die typischerweise zur Erkennung und Isolation der Ausfallsquelle erforderlich sind, und die komplexen Mechanismen und Vorgänge, die typischerweise zur Koordination, Synchronisation und Verwaltung potentiell miteinander kollidierender Fehlerverwaltungsvorgänge notwendig sind.

Detaillierte Beschreibung eines HAN File Servers **10** (**Fig.** 1):

[0021] Es wird Bezug genommen auf **Fig.** 1, in der eine schematische Darstellung eines als Beispiel dienenden HAN File Servers **10** gezeigt ist, in dem die vorliegende Erfindung realisiert werden kann, wie beispielsweise ein Data General Corporation CLARiiON™-Dateiserver. Wie dargestellt, weist ein HAN File Server **10** ein Speicher-Subsystem **12** und ein Steuerungs-Prozessor-Subsystem **14** auf, das aus doppelten Rechenblattelementen/Compute Blades (Blades) **14A** und **14B** besteht, die das Speicher-Subsystem **12** gemeinsam nutzen. Die Rechenblattelemente **14A** und **14B** arbeiten unabhängig, um Clients des HAN File Servers **10** Netzwerkzugriff und Dateisystemfunktionen bereitzustellen und diese zu unterstützen, und zusammenwirkend, um eine gegenseitige Datensicherung und Unterstützung für den Netzwerkzugriff bereitzustellen und den Netzwerkzugriff und die Dateisystemfunktionen jeweils des anderen zu unterstützen.

Speicher-Subsystem **12** (**Fig.** 1):

[0022] Das Speicher-Subsystem **12** weist eine aus einer Vielzahl von Festplattenlaufwerken **18** bestehende Laufwerkbank **16** auf, wobei auf jedes von ihnen bidirektionale Lese-Schreibzugriffe über Doppelspeicher-Schleifenmodule **20** erfolgen, die als Speicher-Schleifenmodule **20A** und **20B** angegeben sind. Wie dargestellt, weisen die Speicher-Schleifenmodule **20A** und **20B** jeweils eine Multiplexerbank (MUXBANK) **22** auf, die als MUXBANKs **22A** und **22B** bezeichnet ist, von denen jede eine Vielzahl von Multiplexern (MUXs) **24** und einen Schleifencontroller **26** aufweist, der jeweils als Schleifencontroller **26A** und **26B** dargestellt ist. Die MUXs **24** und Schleifencontroller **26** eines jeden Schleifencontrollermoduls **20** sind bidirektional durch einen MUX-Schleifenbus **28** miteinander verbunden, der als MUX-Schleifenbusse **28A** und **28B** dargestellt ist.

[0023] Wie dargestellt, weisen die MUXBANKs **22A** und **22B** jeweils einen Plattenlaufwerks-Multiplexer MUX **24** (MUX, **24D**) auf, der mit einem entsprechenden unter den Plattenlaufwerken **18** verbunden ist und diesem entspricht, so daß jedes Plattenlaufwerk **18** der Laufwerkbank **16** eine bidirektionale Lese-Schreibverbindung zu einem entsprechenden DMUX **24D** in jeder der MUXBANKs **20A** und **20B** aufweist. Jede der MUXBANKs **20A** und **20B** ist weiterhin bidirektional mit dem entsprechenden der Rechenblattelemente **14A** und **14B** jeweils durch MUX **24CA** und MUX **24CB** verbunden, wobei die Rechenblattelemente **14A** und **14B** bidirektional durch den Blattelementbus **30** verbunden sind. Zusätzlich kann jede der MUXBANKs **20A** und **20B** einen Externen Plattenanordnungs-Multiplexer MUX **24** umfassen, der als MUXs **24EA** und **24EB** dargestellt, und bidirektional mit dem entsprechenden MUX-Schleifenbus **28A** und **28B** und bidirektional mit einem externen Plattenanordnungs-Multiplexer MUX (EDISKA) **32** verbunden ist, der jeweils als EDISKAs **32A** und **32B** angegeben ist und zusätzlichen oder alternativen Plattenspeicherplatz bereitstellt.

[0024] Daher kommuniziert jedes der Plattenlaufwerke **18** bidirektional mit einem MUX **24** der MUX Bank **22A** und mit einem MUX **24** der MUX Bank **22B**, wobei die MUXs **24** von MUX Bank **20A** über einen Schleifenbus **26A** miteinander verbunden sind, während die MUXs **24** von MUX Bank **22B** über einen Schleifenbus **26B** miteinander verbunden sind, so daß jedes Plattenlaufwerk **18** sowohl über Schleifenbus **26A** als auch über Schleifenbus **26B** zugänglich ist. Zusätzlich kommuniziert das Prozessorblatt **14A** bidirektional mit Schleifenbus **26A**, während Prozessorblatt **14B** bidirektional mit Schleifenbus **26B**, kommuniziert, wobei die Blattelementprozessoren **14A** und **14B** direkt miteinander verbunden sind und über den Blattschleifenelement (Blade) -Bus **30** kommunizieren. Als solche können die Blattelementprozessoren **14A** und **14B** bidirektional mit einem beliebigen der Plattenlaufwerke **18** kommunizieren, entweder direkt über den ihnen zugeordneten Schleifen-

bus **26** oder indirekt über das andere der Blattelementprozessoren **14**, und können direkt miteinander kommunizieren.

[0025] Schließlich handelt es sich, hinsichtlich des Speicher-Subsystems **12** bei der bevorzugten Ausführung eines HAN File Servers **10**, und beispielsweise, bei jedem Plattenlaufwerk **18**, um ein Hot-Swap-Faserkanal-Plattenlaufwerk, das zwecks leichtem Austausch durch den Benutzer in einem Träger eingebaut ist, wobei die Laufwerke und Träger in einer mittleren Ebene eingesteckt sind, die den Strom verteilt und die MUX-Schleifenbusse **26A** und **26B** enthält und dadurch jedes Laufwerk mit doppeltem Port mit den MUXs **24** bzw. die MUXs **24** mit den Schleifencontrollern **26** verbindet. Bei den MUXs **24** handelt es sich um Faserkanal-MUX-Vorrichtungen, wobei die Schleifencontroller **26** Mikrocontroller aufweisen, um die Pfadauswahl einer jeden MUX-Vorrichtung zu steuern, um die doppelten Ports eines jeden Plattenlaufwerkes **18** selektiv mit dem Eingang oder Ausgang der Faserkanal-MUX-Schleifenbusse **26A** und **26B** zu verbinden. Bei den MUXs **24CA** und **24CB** und MUXs **24EA** und **24EB** handelt es sich um ähnliche Faserkanal-MUX-Vorrichtungen, die das Speicher-Subsystem **12** über Faserkanal-Schleifenbusse mit den Rechenblattelementen **14A** und **14B** und den EDISKAs **32A** und **32B** verbinden, während es sich bei dem Rechenblattbus **30** ebenfalls um einen Faserkanalbus handelt.

Steuerungs-Prozessor-Subsystem **14** (**Fig.** 1 und 2):

[0026] Wie weiter oben beschrieben, besteht das Steuerungs-Prozessor-Subsystem **14** aus doppelten Rechenblattelementen (Blades) **14A** und **14B**, die über den Rechenblattelementbus **30** miteinander verbunden sind, und die gemeinsam ein Rechen- und Steuerungs-Subsystem umfassen, das die Vorgänge des gemeinsam genutzten Speicher-Subsystems **12** steuert. Die Rechenblattelemente **14A** und **14B** arbeiten unabhängig, um Clients des HAN File Servers **10** Netzwerkzugriff und Dateisystemfunktionen bereitzustellen und diese zu unterstützen, und zusammenwirkend, um eine gegenseitige Datensicherung und Unterstützung für den Zugriff auf Netzwerk **34** sowie die Dateisystemfunktionen jeweils des anderen zu bieten. Wie in **Fig.** 1 und 2 dargestellt, weist jedes Blade **14** eine Anzahl von mit den Netzwerken **34** verbundenen Netzwerkports (Ports) **34P** auf, die die bidirektionalen Datenkommunikationsverbindungen zwischen dem HAN File Server **10** und den Clients **34C** umfassen, die den HAN File Server **10** benutzen. Wie dargestellt, können die Netzwerke beispielsweise eine Vielzahl von mit den Clients **34C** verbundenen Client-Netzwerken **34N** umfassen, und können auch einen mit entfernten Clients **34C** verbundenen Router **34R** umfassen. Wie von gewöhnlichen Fachleuten auf diesem Fachgebiet verstanden werden wird, können die Netzwerke **34** beispielsweise aus Local Area Networks (LANs), Wide Area Networks (WANs), direkten Prozessorverbindungen oder Bussen, Faseroptikverbindungen oder einer Kombination aus denselben bestehen.

[0027] Wie in **Fig.** 2 angegeben, besteht jedes der Blades **14** aus doppelten Verarbeitungseinheiten **36A** und **36B**, die einen kohärenten Zugriff auf den Speicher und andere Elemente, wie beispielsweise die Verbindungskomponenten, gemeinsam nutzen. Bei jeder der Verarbeitungseinheiten **36A** und **36B** handelt es sich um eine voll funktionsfähige Rechnerverarbeitungseinheit, die einen vollständigen Betriebssystemkern ausführen und in einer funktionellen Mehrfachverarbeitungsstruktur zusammenwirken. Beispielsweise, und wie weiterhin nachfolgend in der Beschreibung der gegenwärtig bevorzugten Realisierung beschrieben wird, führt eine der Verarbeitungseinheiten **36** RAID-Funktionen aus, während die andere Verarbeitungseinheit **36** Netzwerkfunktionen, Protokollprofilfunktionen, CIFS- und NFS-Funktionen, und Dateisystemfunktionen ausführt.

Allgemeiner Aufbau eines HAN File Servers **10** und der Fehlerbehandlungsmechanismen des HAN File Servers **10** (**Fig.** 1 und 2):

[0028] Wie beschrieben, umfaßt zu diesem Zweck ein HAN File Server **10** der vorliegenden Erfindung ein Cluster aus hierarchischen und gleichrangigen Domänen, d.h. Knoten- oder Subsysteme, wobei jede Domäne eine oder mehrere Aufgaben oder Funktionen des Dateiservers ausführt und Fehlerbehandlungsmechanismen umfaßt. Der HAN File Server **10** besteht beispielsweise aus drei hierarchischen Domänen **10A**, **10B** und **10C**, die jeweils Netzwerke **34N**, ein Steuerungs-Prozessor-Subsystem **14** und ein Speicher-Subsystem **12** umfassen, die separate und komplementäre Funktionen des Dateiservers ausführen. Dies bedeutet, daß die Domäne **10A** Client-Serververbindungen zwischen Clients **34** und dem HAN File Server **10** bereitstellt, Domäne **10B**, d.h. das Steuerungs-Prozessor-Subsystem **14**, die Client-Serververbindungen von Domäne **10A** und übergeordnete Dateisystemtransaktionen unterstützt, und Domäne **10C**, d.h. Speicher-Subsystem **12**, die Dateisysteme der Clients unterstützt. Das Steuerungs-Prozessor-Subsystem **14** besteht wiederum aus zwei gleichrangigen Domänen **10D** und **10E**, d.h. Blades **14A** und **14B**, die parallele Funktionen ausführen, insbesondere Client-Serverkommunikationsfunktionen und übergeordnete und untergeordnete Dateisystemvorgänge, wodurch sie die Clientkommunikationen und Dateivorgangs-Aufgabenlasten teilen. Wie in der nachfolgenden Beschreibung detailliert beschrieben wird, umfassen die die Blades **14A** und **14B** enthaltenden Domänen auch unabhängig arbeitende Fehlerbehandlungsmechanismen, die eine Fehlerbehandlung und Unterstützung für Client-Serverkommunikationen, Kommunikationen zwi-

schen den Blades **14**, übergeordnete Dateisystemfunktionen und im Speicher-Subsystem **12** ausgeführte untergeordnete Dateisystemfunktionen bereitstellen. Bei jedem Blade **14** handelt es sich wiederum um eine aus zwei hierarchischen Domänen **10F** und **10G** bestehende Domäne, die auf den Verarbeitungseinheiten **36A** und **36B** basiert, die separate, jedoch komplementäre Funktionen ausführen, die gemeinsam die Funktionen der Blades **14A** und **14B** darstellen. Wie beschrieben werden wird, bildet eine der Verarbeitungseinheiten **36** die obere Domäne **10F**, die übergeordnete Dateivorgänge und Client-Serverkommunikationen mit Fehlerbehandlungsmechanismen für beide Funktionen bereitstellt. Die andere der Verarbeitungseinheiten **36** bildet die untere Domäne **10G**, die untergeordnete Dateivorgänge und Kommunikationen zwischen den Blades **14** mit unabhängig arbeitenden Fehlerbehandlungsmechanismen bereitstellt, die beide Funktionen und die Serverfunktionen und Fehlerbehandlungsmechanismen der oberen Domäne **10F** unterstützen. Schließlich besteht das Speicher-Subsystem **12** auf ähnliche Art und Weise aus einer unteren Domäne **10H**, die die Plattenlaufwerke **18** umfaßt, d.h. die Speicherelemente des Servers, und die indirekt die RAID-Mechanismen unterstützt, die von den Domänen **10E** der Blades **14** unterstützt werden, und die oberen gleichrangigen Domänen **10I** und **10J**, die Speicher-Schleifenmodule **20A** und **20B** umfassen, die Kommunikationen zwischen den Domänen **10D** und **10E** und der Domäne **10H** unterstützen.

[0029] Daher, und wie nachfolgend beschrieben wird, enthält bzw. umfaßt jede Domäne des HAN File Servers **10** direkt oder indirekt einen oder mehrere Fehlerbehandlungsmechanismen, die voneinander unabhängig und separat, jedoch ohne einen einzelnen, zentralen Master- oder Koordinationsmechanismus miteinander zusammenwirkend arbeiten, so daß die Funktionen oder Vorgänge einer ausgefallenen Komponente einer Domäne von einer entsprechenden Komponente einer damit in Zusammenhang stehenden Domäne übernommen werden. Zusätzlich, und wie ebenfalls nachfolgend beschrieben wird, verwenden bestimmte unter den Fehlerbehandlungsmechanismen eines HAN File Servers **10** viele verschiedene Technologien oder Verfahren auf transparente Art und Weise, um bei einzelnen oder mehrfachen Ausfällen eine fortgesetzte Funktionsfähigkeit bereitzustellen.

[0030] Nach nun erfolgter Beschreibung der Gesamtstruktur und des Betriebes eines HAN File Servers **10** wird nachfolgend jede Domäne eines HAN File Servers **10** noch detaillierter beschrieben, sowie die Struktur und der Betrieb der Fehlerbehandlungsmechanismen des HAN File Servers **10**.

Verarbeitungs- und Steuerungskern eines Blade **14**:

[0031] Es wird Bezug genommen auf **Fig.** 2, in der eine gegenwärtig bevorzugte Realisierung eines Blade **14** veranschaulicht ist, wobei dargestellt ist, daß ein Blade **14** doppelte Prozessoren **38A** und **38B** aufweist, die jeweils die Rechenkerne der doppelten Verarbeitungseinheiten **36A** und **36B**, und eine Anzahl gemeinsam genutzter Elemente bilden, wie beispielsweise den Speicher-Controller-Hub (MCH) **38C**, Speicher **38D** und einen Eingabe-Ausgabe-Controller-Hub (ICH) **38E**. Bei einer gegenwärtigen Realisierung handelt es sich beispielsweise bei jedem der Prozessoren **38A** und **38B** um einen Intel Pentium-III-Prozessor mit einem internen Level 2-Pufferspeicher, wobei es sich bei MCH **38C** und ICH **38E** um einen Intel 820-Chipsatz handelt und Speicher **38D** aus 512 MB RDRAM oder SDRAM besteht, aber auch größer sein kann.

[0032] Wie dargestellt, sind die Prozessoren **38A** und **38B** mit MCH **38C** über einen mittels Pipelining ausgebildeten Front Side Bus (FSB) **38F** und einen entsprechenden FSB-Port **38Ca** von MCH **38C** miteinander verbunden. Wie von gewöhnlichen Fachleuten auf diesem Fachgebiet gut verstanden werden wird, unterstützt der FSB-Port von MCH **38C** und MCH **39C** die Initiierung und den Empfang von Speicherreferenzen von den Prozessoren **38A** und **38B**, die Initiierung und den Empfang von Eingabe-Ausgabe-(I/O)- und speicherabgebildeten I/O-Anforderungen von den Prozessoren **38A** und **38B**, die Lieferung von Speicherdaten von Speicher **38C** zu den Prozessoren **38A** und **38B**, und die Initiierung von Speicher-Schnüffelzyklen, die das Ergebnis von Speicher-I/O-Anforderungen sind. Wie ebenfalls dargestellt ist, umfaßt MCH **38C** weiterhin einen Speicher-Port **38Cb** zu Speicher **38D**, einen Hublink-Port **38Cc**, der mit einem Hublink-Bus **38G** bzw. mit ICH **38E** verbunden ist, und vier AGP-Ports **38Cd**, die als Personal Computer Interconnect (PCI)-Busse nach Industriestandard arbeiten, von denen jeder mit einem Prozessor bzw. mit Prozessorbrückeneinheit (P-P-Bridge) **38H**, wie beispielsweise einem Intel 21154-Chip, verbunden ist.

[0033] Der ICH **38E** weist wiederum einen Hublink-Port **38Ea** auf, der mit dem Hublink-Bus **38G** bzw. mit MCH **38C** verbunden ist, einen mit einem Firmware-Speicher **38I** verbundenen Firmware-Port **38Eb**, einen mit einem Hardware-Monitor (HM) **38J** verbundenen MonitorPort **38Ec**, und einen mit einem Boot-Laufwerk **38K** verbundenen IDE-Laufwerks-Port **38Ed**, einen mit einer Super I/O-Vorrichtung (Super I/O) **38L** verbundenen I/O-Port **38Ee**, und einen PCI-Port **38Ef**, der, zusätzlich zu anderen Elementen, auch mit einer VGA-Vorrichtung (VGA) **38M** und einer Local Area Network-Verwaltungsvorrichtung (LAN) **38N** verbunden ist, wobei dies alles von gewöhnlichen Fachleuten auf diesem Fachge-

biet gut verstanden werden wird.

PC-Kompatibilitäts-Subsystem eines Blade **14**:

[0034] ICH **38E**, Super I/O **38L** und VGA **38M** umfassen gemeinsam ein PC-Kompatibilitäts-Subsystem (PC), das PC-Funktionen und -Dienste für den HAN File Server **10** zum Zwecke der lokalen Steuerung und von Anzeigefunktionen bereitstellt. Zu diesem Zweck stellt der ICH **38E**, was von gewöhnlichen Fachleuten auf diesem Fachgebiet gut verstanden werden wird, IDE-Controller-Funktionen, einen IO APIC, auf 82C59-basierende Zeitgeber und eine Echtzeituhr bereit. Bei Super IO **38L** kann es sich wiederum beispielsweise um ein Standard Microsystems Device LPC47B27x handeln, das einen 8042 Tastatur-Maus-Controller, einen 2,88 MB-Super I/O-Diskettenlaufwerks-Controller und voll funktionsfähige serielle Ports bereitstellt, während es sich bei VGA **38M** beispielsweise um einen Cirrus Logic 64-Bit VisualMedia® Accelerator CL-GD5446-QC handeln kann, der einen 1 MB-Rahmenpufferspeicher unterstützt.

Firmware und BIOS-Subsystem eines Blade **14**:

[0035] Der ICH **38E** und Firmware-Speicher **38I** umfassen gemeinsam ein Firmware- und BIOS-Subsystem, das die üblichen Firmware- und BIOS-Funktionen ausführt, einschließlich Selbsttest beim Einschalten (POST) und einer vollständigen Konfiguration von Ressourcen des Blade **14A** und **14B**. Die Firmware und BIOS, wobei es sich beispielsweise um ein Standard-BIOS handeln kann, wie es von AMI/Phoenix, erhältlich ist, sind im Firmware-Speicher **38I** resident, der einen 1 MB Flash-Speicher enthält. Nach Abschluß des POST fragt das BIOS die oben beschriebenen PCI-Busse ab, und konfiguriert während dieser Abfrage die zwei oben beschriebenen und in der nachfolgenden Beschreibung beschriebenen PCI-PCI-Brücken, und erkennt das Vorhandensein des Faserkanals und der LAN-Controller auf den Back-End- und Front-End-PCI-Bussen, die in einer nachfolgenden Abhandlung beschrieben werden, und bildet sie im PCI-Adreßbereich ab. Diese Informationen werden in MP-anpaßbaren Tabellen notiert, die die Topologie des IO-Subsystems mit den anderen Standard-Größeneinteilungsinformationen beschreiben, wie beispielsweise IO-PC-Kompatibilität, Speichergröße usw., wobei POST eine einfache Pfadüberprüfung und Speicherdiagnose durchführt. Nach Abschluß des POST wird ein im Flash-Speicher residentes Benutzer-Binärcodesegment geladen, das ein tiefgehendes Vor-Boot-Diagnosepaket umfaßt, das auch die Faserkanalvorrichtungen initialisiert und die Integrität der Komponenten auf dem Rechenblattelement durch Anwendung von musterempfindlichen Daten auf Datenpfade und DRAM-Zellen überprüft. Nach Ausführung der Diagnose kehrt die Kontrolle entweder zum BIOS oder zu einem Startroutinen-Dienstprogramm zurück. Wenn die Kontrolle dem BIOS übergeben wird, fährt das System mit dem Booten fort, und wenn die Kontrolle dem Startroutinen-Dienstprogramm übergeben wird, wird der Boot-Block aus der Faserplatte gelesen, und die Kontrolle wird dem Bild des neu geladenen Betriebssystems übergeben. Zusätzlich stellt dieses Subsystem Merkmale und Funktionen zur Unterstützung des Aufbaus der Systemverwaltung insgesamt bereit, einschließlich Fehlerprüfungslogik, Umgebungsüberwachung sowie Fehler- und Schwellenwertprotokollierung. Auf dem niedrigsten Niveau werden Hardwarefehler- und Umgebungs-Schwellenwertüberprüfungen ausgeführt, die interne Prozessor-Pufferspeicher-Paritäts-ECC-Fehler, PCI-Bus-Paritätsfehler, RDRAM-ECC-Fehler und Front Side Bus-ECC-Fehler beinhalten. Fehler- und Ereignisse mit überschrittenen Umgebungsschwellenwerten werden in einem Abschnitt des Flash-PROMS in einem DMI-kompatiblen Aufzeichnungsformat protokolliert.

I/O-Bus-Subsysteme eines Blade **14**:

[0036] Schließlich unterstützen MCH **38C** und ICH **3E** zwei Blade 14-Input-Output (I/O)-Bus-Subsysteme, wobei es sich bei dem ersten um ein Back-End Bus-Subsystem (BE Bus-Sys) **38O** handelt, das von MCH **38C** unterstützt wird und die zuvor beschriebenen bidirektionalen Verbindungen zwischen dem Blade **14** und dem entsprechenden Schleifenbus **26** des Speicher-Subsystems **12** und die bidirektionale Verbindung zwischen den Blades **14A** und **14B** über Rechenblattelementbus **30** bereitstellt. Bei dem zweiten handelt es sich um ein Front-End Bus-Subsystem (FE BusSys) **38P**, das durch den ICH **38E** unterstützt wird, der die zuvor beschriebenen bidirektionalen Verbindungen zu und von den Netzwerken **34** bereitstellt, wobei die Netzwerke **34**, wie zuvor abgehandelt, beispielsweise aus Local Area Networks (LANs), Wide Area Networks (WANs), direkten Prozessorverbindungen oder Bussen, Faseroptikverbindungen oder einer beliebigen Kombination aus denselben bestehen können.

[0037] Wenn wir zuerst BE BusSys **38O** betrachten, unterstützt MCH **38C**, wie weiter oben beschrieben, vier AGP-Ports **38Cd**, die als Personal Computer Interconnect (PCI)-Busse nach Industriestandard arbeiten. Jeder AGP-Port **38Cd** ist mit einer Prozessor-Prozessor-Brückeneinheit (P-P-Bridge) **38H** verbunden, wie beispielsweise einem Intel 21154-Chip, der wiederum mit den bidirektionalen Bus-Ports von zwei Faserkanal-Controllern (FCCs) **38Q** verbunden ist, die beispielsweise Tach Lite-Faserkanal-Controller umfassen können. Die parallelen Faserkanalschnittstellen der FCCs **38Q** sind wiederum mit den parallelen Faserkanalschnittstellen von zwei entsprechenden Serialisierungs-Deserialisierungsvorrichtungen (SER-DES) **38R** verbunden. Die serielle Schnittstelle eines SER-DES **38R** ist mit dem Re-

chenblattelementbus **30** verbunden, um die Kommunikationsverbindung zu dem anderen der doppelten Blades **14** bereitzustellen, während die serielle Schnittstelle des anderen SER-DES **38R** mit dem entsprechenden Schleifenbus **26** des Speicher-Subsystems **12** verbunden ist.

[0038] Im FE BusSys **38P**, und wie weiter oben beschrieben, umfaßt der ICH **38E** einen PCI-Port **38Ef** und, wie dargestellt, ist der PCI-Port **38Ef** bidirektional zu einer PCI-Bus-PCI-Bus-Brückeneinheit (P-P Bridge) **385**, die beispielsweise aus einem Intel 21152-Element bestehen kann, das ein bidirektionales 32 Bit, 33 MHz-Front-End-PCI-Bussegment unterstützt. Das Front-End-PCI-Bussegment ist wiederum mit einem Set bidirektionaler Netzwerkvorrichtungen (NETDEVs) **38T** verbunden, die mit den Netzwerken **34** verbunden sind, und bei denen es sich beispielsweise um Intel 82559 10/100 Ethernet-Controllervorrichtungen handeln kann. Es wird verstanden werden, daß Netzwerke **34**, wie zuvor beschrieben wurde, beispielsweise aus Local Area Networks (LANs), Wide Area Networks (WANs), direkten Prozessorverbindungen oder Bussen, Faseroptikverbindungen oder einer Kombination aus denselben bestehen können, und die NETDEVs **38T** dementsprechend ausgewählt werden.

[0039] Schließlich sollte im Hinblick auf BE BusSys **38O** und FE BusSys **38P** bemerkt werden, daß es sich sowohl beim BE BusSys **38O** als auch beim FE BusSys **38P** bei der gegenwärtig bevorzugten Ausführung um PCI-Busse handelt, die als solche eine Struktur eines Gleichtakt-Unterbrechers aufweisen. Aus diesem Grund sind die PCI-Unterbrecher von BE BusSys **38O** und FE BusSys **38P** so geleitet, daß die PCI-Busvorrichtungen von BE BusSys **38O** keine Unterbrecher gemeinsam mit den PCI-Busvorrichtungen von FE BusSys **38P** nutzen.

Betrieb eines HAN File Servers **10** (**Fig.** 1, 2, 3 und 4):

Allgemeiner Betrieb eines HAN-Dateisystems **10**:

[0040] Wie zuvor beschrieben, umfaßt ein HAN-Dateisystem **10** doppelte Rechenblattelemente **14**, von denen jedes über einen kompletten Zugriff auf alle Plattenlaufwerke **18** des Speicher-Subsystems **12**, und Verbindungen zu allen Client-Netzwerken **34N** verfügt, von denen jedes für sich selbst dazu in der Lage ist, alle Funktionen und Vorgänge des HAN-Dateisystems **10** auszuführen. Eine schematische Darstellung der funktionellen und betriebsmäßigen Struktur eines Blade **14** ist in **Fig.** 3 veranschaulicht. **Fig.** 3 stellt ein einzelnes Blade der Blades **14A** und **14B** dar, und es wird verstanden werden, daß das andere der Blades **14** mit dem veranschaulichten Blade **14** identisch, und ein Spiegelbild desselben ist.

[0041] Innerhalb eines Blade **14**, und wie weiter oben beschrieben, nutzen die doppelten Verarbeitungseinheiten **36A** und **36B** gemeinsam eine Anzahl von Blade 14-Elementen, wie beispielsweise einen Speicher-Controller-Hub (MCH) **38C**, einen Speicher **38D** und einen Eingabe-Ausgabe-Controller-Hub (ICH) **38E**. Die Verarbeitungseinheiten **36A** und **36B** arbeiten jeweils unabhängig, jedoch mit der jeweils anderen zusammenwirkend, wobei jede eine separate Kopie eines in Speicher **38A** residenten Echtzeit-Betriebssystems/Operating System (OS) **40** ausführt, und jede Kopie des OS **40** beispielsweise eine grundlegende Speicherverwaltung, Aufgabenzeitplanung und Synchronisationsfunktionen und andere grundlegende Betriebsystemfunktionen für die entsprechende der Verarbeitungseinheiten **36A** und **36B** bereitstellt. Die Verarbeitungseinheiten **36A** und **36B** kommunizieren über einen in dem gemeinsam genutzten Speicher **38A** realisierten Message-Weiterleitungsmechanismus (Message) **42** miteinander, wobei Meldungen beispielsweise zum Starten eines I/O-Vorgangs, zum Abschluß eines I/O-Vorgangs, zur Ereignisbenachrichtigung, wie beispielsweise über einen Plattenausfall, für Statusabfragen und zur Spiegelung von kritischen Datenstrukturen wie beispielsweise des Dateisystemprotokolls, das durch den Blattelementbus **30** gespiegelt wird, definiert werden. Bei der Initialisierung lädt jedes Blade **14** beide Kopien des OS **40** und die RAID-, Dateisystem- und Netzwerkabbildungen von den Back-End-Plattenlaufwerken **18**. Die zwei RAID-Kernels, von denen jeder in eine der Verarbeitungseinheiten **36A** und **36B** ausführt, teilen dann zusammenwirkend den Speicher **38A** des Blade **14** zwischen den zwei Instanzen des OS **40** auf, und initiieren Vorgänge der Verarbeitungseinheiten **36A** und **36B** nach dem Laden der Kopien des OS 40-Kernels. Nach der Initialisierung kommunizieren die OS 40-Kernel über Message **42**.

[0042] Wie in **Fig.** 3 veranschaulicht, ist innerhalb eines jeden Blade **14** eine der Verarbeitungseinheiten **36A** und **36B** als Back-End-Prozessor (BEP) **44B** bestimmt, und arbeitet auch als solcher, und arbeitet, wie weiter oben beschrieben, als ein Blockspeichersystem zum Schreiben und Lesen von Daten zu und von RAID-Konfigurationsplatten und weist einen RAID-Mechanismus (RAID) **46** mit einem RAID-Dateimechanismus (RAIDF) **46F** auf, der RAID-Datenspeicherungs- und Datensicherungsfunktionen ausführt, sowie einen RAID-Monitor-Mechanismus/RAID Monitor Mechanismus (RAIDM) **46M**, der mit RAID in Zusammenhang stehende Systemüberwachungsfunktionen sowie andere, unten beschriebene Funktionen ausführt. Die andere der Verarbeitungseinheiten **36A** und **36B** ist als Front-End-Prozessor (FEP) **44F** bestimmt, und arbeitet auch als solcher, und führt alle Netzwerk- und Dateisystemvorgänge zur Übertragung von Daten zwischen den Clients und dem plattenspeicherresidenten Blockspeichersystem

und dazugehörige RAID-Funktionen des BEP **44B** aus, einschließlich Unterstützung der Netzwerktreiber, Protokollprofile, einschließlich CIFS- und NFS-Protokollen und der Aufrechterhaltung eines Journaled File Systems.

[0043] Zusätzlich zu Blockspeichersystemvorgängen umfassen die Funktionen des BEP **44B** die Ausführung von Kern-RAID-Dateisystem-Unterstützungsalgorithmen über RAIDF **46F** und, über RAIDM **46M** die Überwachung des Betriebes der Plattenlaufwerke **18**, Überwachung der Vorgänge und des Status beider Blades **14**, in denen er resident ist, und des gleichrangigen Blade **14**, sowie das Berichten über Ausfälle an die verwaltenden Funktionen. Wie weiter oben in bezug auf **Fig.** 2 und BE BusSys **38O** beschrieben, unterstützt der BEP **44B** auch Kommunikationen zwischen den Blades **14A** und **14B** über BE BusSys **38O** und Blattelementbus **30** sowie mit den Plattenlaufwerken **18** über BE BusSys **38O** und den entsprechenden Schleifenbus **26** des Speicher-Subsystems **12**. RAIDM **46M** überwacht ebenfalls die Stromversorgung von Blade **14** und führt bei einem Stromausfall angemessene Aktionen aus, wie beispielsweise die Ausführung eines Notschreibvorgangs kritischer Datenstrukturen auf die Plattenlaufwerke **18** und Benachrichtigung der anderen der Verarbeitungseinheiten **36A** und **36B**, so daß die andere der Verarbeitungseinheiten **36A** und **36B** eine angemessene Aktion einleiten kann. Der BEP **44B** stellt weiterhin bestimmte Startroutinen-Unterstützungsfunktionen bereit, wobei Laufzeitkernels auf den Plattenlaufwerken **18** gespeichert, und beim Booten des Systems geladen werden können.

[0044] Der FEP **44F** weist wiederum Netzwerkmechanismen (Network) **48** auf, die alle mit Netzwerk **34** in Zusammenhang stehenden Funktionen und Vorgänge des Blade **14** ausführen, und die Elemente des FE BusSys **30P** und NetDevs **38T** umfaßt. Beispielsweise verwaltet Network **48** die für Netzwerk-Clients verfügbaren Ressourcen und stellt sie bereit, einschließlich FE BusSys **38P**, um Clients **34C** über die Netzwerke **34** Zugriff auf das HAN-Dateisystem **10** zu bieten. Wie beschrieben werden wird, unterstützt Network **48** auch in dem FEP **44F** residente Kommunikations-Failover-Mechanismen und andere Hochverfügbarkeitsmerkmale, wie in diesem Dokument beschrieben.

[0045] Der FEP **44F** weist auch ein Journaled File System (JFile) **50** auf, das mit Clients des HAN File Servers **10** über Network **48**, und mit den RAID-Dateisystemfunktionen von RAIDF **46F** über Message **42** kommuniziert. Wie angegeben, weist JFile **50** einen Dateisystem-Mechanismus (FSM) **50F** auf, der die Dateisystemfunktionen von JFile **50** und einen internen Schreib-Pufferspeicher (WCache) **50C** und ein Transaktionsprotokoll (Log) **50L** ausführt, die mit FSM **50F** zusammenarbeiten, um jeweils die Daten und Vorgänge von Datentransaktionen zu puffern und eine Aufzeichnung von Datentransaktionen aufrechtzuerhalten. Log **50L** weist wiederum einen Protokollerzeuger (LGen) **50G** zum Erzeugen von Protokolleinträgen (SEs) **50E** auf, die angeforderte Datentransaktionen darstellen sowie einen Protokollspeicher (LogM) **50M** zum Speichern von SEs **50E**, wobei die Tiefe des LogM **50M** von der Anzahl von aufzuzeichnenden Datentransaktionen abhängig ist, wie weiter unten abgehandelt werden wird. Wie angegeben, weist der BEP **44B** einen Pufferspeicher-Spiegelungsmechanismus/Cache Mirror Mechanism (Cmirror) **54C** auf, der mit WCache **50C** kommuniziert und die Inhalte von WCache **50C** spiegelt. Zusätzlich wird das Log **50L** eines jeden Blade **14** durch einen Protokoll-Spiegelungsmechanismus/Log Mirror Mechanism (LMirror) **54L** für Log **50L** gespiegelt, das in dem entgegengesetzten gleichrangigen Blade **14** resident ist, wobei das Log **50L** eines jeden Blade **14** mit dem entsprechenden LMirror **54L** über den Pfad kommuniziert, der Message **42**, BE BusSys **38O** und Blattelementbus **30** umfaßt.

[0046] Schließlich weist der FEP **44F** einen Statusüberwachungsmechanismus (Monitor) **52** auf, der Benachrichtigungen vom BEP **44B** in bezug auf Änderungen im HAN-Dateisystem **10** überwacht und angemessene Aktionen als Reaktion auf diese Änderungen initiiert. Diese Benachrichtigungen können beispielsweise Benachrichtigungen von RAIDM **46M** in bezug auf die Einbindung neu eingebauter Platten in eine RAID-Gruppe, oder den Aufbau einer SNMP-Auffangroutine für eine ausgefallene Platte sein, und die durch Monitor **52** initiierten Vorgänge können beispielsweise die Initiierung eines Failover-Vorgangs oder der vollständigen Abschaltung des Blade **14** durch die Fehlerbehandlungsmechanismen des HAN File Servers **10** umfassen, wie nachfolgend beschrieben werden wird, wenn die RAID-Funktionen einen ausreichend schweren Fehler entdecken, usw.

Betrieb der Dateisystemmechanismen eines HAN File Servers **10** (**Fig.** 1, 2 und 3):

[0047] Wie weiter oben in diesem Dokument beschrieben, und wie in **Fig.** 3 veranschaulicht, umfassen die Dateisystemmechanismen eines HAN File Servers **10** drei primäre Komponenten oder Schichten, wobei es sich bei der ersten und obersten Schicht um die Dateisystemmechanismen von JFile **50** mit WCache **50C** und Log **50L** handelt, die auf den Front-End-Prozessoren **44F** jedes der Blades **14A** und **14B** resident sind. Die unterste Schicht umfaßt das Speicher-Subsystem **12** mit den Plattenlaufwerken **18** und den Blockspeicherfunktionen und RAIDF 46F-Funktionen, die auf den BEPs **44B** eines jeden der Blades **14A** und **14B** resident sind. Die dritte Schicht oder Komponente der Dateisystemmechanismen des HAN File Servers **10** bestehen aus den

Fehlerbehandlungsmechanismen zur Erkennung und Behandlung von Fehlern, die den Betrieb der Dateisystemmechanismen beeinträchtigen, und zur Wiederherstellung nach Dateisystemausfällen. Die Struktur und der Betrieb der oberen und unteren Dateisystemelemente wurden oben abgehandelt und beschrieben, sind den bekannten ähnlich, und werden von gewöhnlichen Fachleuten auf diesem Fachgebiet gut verstanden. Als solche werden diese Elemente der beispielhaften Dateimechanismen des HAN File Servers **10** in diesem Dokument nicht detailliert abgehandelt, außer in dem Umfang, in dem dies für ein vollständiges Verständnis der vorliegenden Erfindung notwendig ist. Die nachfolgenden Abhandlungen werden sich stattdessen auf die Fehlerbehandlungsmechanismen des Dateimechanismen des HAN File Servers **10**, und insbesondere auf die in bezug zum Betrieb der Dateisystemelemente des oberen Niveaus des HAN File Servers **10** stehenden Fehlerbehandlungsmechanismen konzentrieren.

[0048] Wie beschrieben, besteht die dritte Komponente der Dateimechanismen des HAN File Servers **10** aus Spiegelungsmechanismen, die einen Schutz gegen den Datenverlust bieten, der aus dem Verlust einer beliebigen Komponente des HAN File Servers **10** resultiert. Wie in **Fig.** 3 veranschaulicht, umfassen die Spiegelungsmechanismen für jedes Blade **14** einen Pufferspeicher-Spiegelungsmechanismus (CMirror) **54C**, der in dem BEP **44B** des Blade **14** resident ist, und einen Protokoll-Spiegelungsmechanismus (LMirror) **54L**, der in dem BEP **40B** des entgegengesetzten gleichrangigen Blade **14** resident ist. Bei CMirror **54C** handelt es sich um einen ununterbrochen arbeitenden Pufferspeicher-Spiegelungsmechanismus, der mit WCache **50C** von JFile **50** über Message **42** kommuniziert. Log **50L** wird wiederum auf Anfrage durch den LMirror **54L** gespiegelt, der in dem BEP **44B** des gleichrangigen Blade **14** resident ist und mit dem entsprechenden LogM **50M** über den Pfad mit Message **42**, BE BusSys **38O** und Rechenblattelementbus **30** kommuniziert, so daß alle Datenänderungen an den Dateisystemen durch eines der Blades **14A** oder **14B** zu dem anderen der Blades **14A** und **14B** reflektiert werden, bevor dem Client gegenüber über diese eine positive Rückmeldung erfolgt. In dieser Hinsicht, und bei der gegenwärtig bevorzugten Ausführung, wird die Spiegelung eines Log **50L** während der Verarbeitung jeder Dateisystemtransaktion ausgeführt, so daß die Latenzzeit der Transaktionsprotokollspiegelung in größtmöglichem Umfang durch die Ausführung der eigentlichen Dateisystemtransaktion maskiert wird. Schlußendlich wird verstanden werden, daß die Plattenlaufwerk 18-Dateisysteme, -Steuerungs-, -Überwachungs- und -Datenwiederherstellungs/-rekonstruktionsfunktionen, die durch RAIDF **46F** unterstützt und bereitgestellt werden, zusätzlich ein Teil der Datenschutzmechanismen des HAN File Servers **10** sind, wobei interne Datenspiegelungsverfahren des Speicher-Subsystems **12** verwendet werden.

[0049] Wie in nachfolgenden Abhandlungen weiter beschrieben werden wird, unterstützen diese Spiegelungsmechanismen daher eine Anzahl alternativer Verfahren zur Behandlung eines Ausfalles in einem Blade **14** in Abhängigkeit von der Ausfallsart. So kann beispielsweise bei einem Ausfall eines Blade **14** das noch funktionsfähige Blade **14** die in seinem LMirror **54L** gespeicherten Dateitransaktionen in das ausgefallene Blade **14** zurücklesen, wenn das ausgefallene Blade **14** wiederhergestellt wurde, so daß es wieder in Betrieb gehen kann, woraufhin alle verlorenen Dateitransaktionen durch das wiederhergestellte Blade **14** erneut ausgeführt und wiederhergestellt werden können. Bei anderen Verfahren, und wie weiterhin in bezug auf die Netzwerk 34-Failover-Mechanismen der Blades **14** beschrieben werden wird, können an das ausgefallene Blade **14** gerichtete Dateitransaktionen erneut auf das noch funktionsfähige Blade **14** umgeleitet werden, und zwar entweder über den Blattelementbus 30-Pfad zwischen den Blades **14** oder durch Umleitung der Clients auf das noch funktionsfähige Blade **14** mittels der Netzwerk 34-Failover-Mechanismen der Blades **14**. Das noch funktionsfähige Blade **14** übernimmt dadurch die Ausführung der an das ausgefallene Blade **14** gerichteten Dateitransaktionen. Wie unten beschrieben, kann das noch funktionsfähige Blade **14** als Teil dieses Vorgangs alle verlorenen Dateitransaktionen des ausgefallenen Blade **14** entweder erneut ausführen oder wiederherstellen, indem es die in seinem LMirror **54L** gespeicherten Dateitransaktionen von dem ausgefallenen Blade **14** erneut ausführt, oder kann die Dateitransaktionen wieder in das ausgefallene Blade **14** zurücklesen, nachdem das ausgefallene Blade **14** wiederhergestellt wurde, wodurch der Zustand des Dateisystems auf dem ausgefallenen Blade **14** zum Zeitpunkt des Ausfalls wiederhergestellt wird, so daß keine Daten für anerkannte Transaktionen aus dem ausgefallenen Blade **14** verlorengehen.

Betrieb der Kommunikationsmechanismen eines HAN File Servers **10** (**Fig.** 1, 2 und 3):

[0050] Wie in **Fig.** 1, 2 und 3 veranschaulicht, können die Kommunikationsmechanismen eines HAN File Servers **10** mit der vorliegenden Erfindung als aus drei Niveaus oder Schichten von Kommunikationsmechanismen bestehend betrachtet werden. Zum Zwecke der vorliegenden Beschreibungen besteht das oberste Niveau aus mit Netzwerk **34** in Zusammenhang stehenden Kommunikationsmechanismen zur Kommunikation von Dateitransaktionen zwischen Clients **34C** und den vom HAN File Server **10** unterstützten Client-Dateisystemstrukturen, und den damit in Zusammenhang stehenden Kommunikationsausfallbehandlungsmechanismen. Die mittlere Schicht der Kommunikationsmechanis-

men umfaßt Kommunikationsmechanismen, die Kommunikationen zwischen den Blades **14A** und **14B** unterstützen, wie beispielsweise Blattelementbus **30** und Messages **42**, und die damit in Zusammenhang stehenden Kommunikationsausfallsmechanismen. Die unterste Schicht der Kommunikationsmechanismen umfaßt die Pfade und Mechanismen zur Kommunikation zwischen den Blades **14** und dem Speicher-Subsystem **12** und zwischen den Elementen des Speicher-Subsystems **12**, was oben abgehandelt wurde und nicht weiter abgehandelt wird, außer in dem Umfang, in dem dies für ein vollständiges Verständnis der vorliegenden Erfindung notwendig ist.

[0051] Wenn wir zuerst das obere Niveau oder Schicht der Kommunikationsmechanismen eines HAN File Servers **10** betrachten, wie in **Fig.** 3 veranschaulicht, dann umfassen die auf dem FEP **44F** eines jeden der Blades **14A** und **14B** residenten Netzwerkmechanismen (Network) **48** ein Netzwerkstapel-Betriebssystem/Network Stack Operating System (NetSOS) **56**, das ein TCP-IP-Protokollprofil (TCP/IP Stack) **58** und einen Netzwerkvorrichtungstreiber/Network Device Drivers (NetDDs) **60** umfaßt, wobei, wie unten beschrieben, diese Mechanismen so erweitert sind, daß sie einzelne Port 34P-Ausfälle, Netzwerk 34-Ausfälle und Ausfälle des gesamten Blade **14** aufnehmen und diese bearbeiten. In dieser Hinsicht, und wie in diesem Dokument woanders abgehandelt, kann das Netzwerk **34** beispielsweise aus Local Area Networks (LANs), Wide Area Networks (WRNs), direkten Prozessorverbindungen oder Bussen, Faseroptikverbindungen oder einer Kombination aus denselben bestehen, wobei die NETDEVs **38T** und NetDDs **60** dementsprechend realisiert werden.

[0052] Wie ebenfalls in **Fig.** 3 veranschaulicht und weiter unten in bezug auf die Hochverfügbarkeitskommunikationsmechanismen eines HAN File Servers **10** abgehandelt, weist jedes Network **48** weiterhin eine Client-Leitwegtabelle/Client Routing Table (CRT) **48A** zum Speichern von Client-Leitwegeinträgen/Client Routing Entries (CREs) **48E** auf, die Leitweg- und Adreßinformationen von Clients **34C** enthalten, die durch Blade **14** unterstützt werden, und CREs **48E** von Clients **34C**, die durch das entgegengesetzte gleichrangige Blade **14** unterstützt werden. Wie von gewöhnlichen Fachleuten auf diesem Fachgebiet verstanden werden wird, können CREs **48E** von Network **48** verwendet werden, um Dateitransaktionskommunikationen an einen vorgegebenen Client **34C** zu richten, und wenn notwendig, Dateitransaktionskommunikationen zu identifizieren oder zu bestätigen, die von diesen einem Blade **14** zugeordneten Clients **34C** empfangen wurden. Wie angegeben, weist jedes Network **48** auch eine Blattelement-Leitwegtabelle/Blade Routing Table (BRT) **48B** auf, die in Zusammenhang mit Kommunikationspfaden von Netzwerk **34** stehende Adreß- und Leitweginformationen umfaßt, die für die Blades **14** zugänglich sind und gemeinsam von diesen genutzt wird, wodurch potentielle Kommunikationspfade zwischen den Blades **14** gebildet werden. Bei einer typischen und gegenwärtig bevorzugten Realisierung von Networks **48**, CRT **48A** und BRT **48B** werden Informationen zwischen den Blades **14A** und **14B** durch den Kommunikationspfad mit Blattelementbus **30** ausgetauscht, können aber jedem Blade **14** beispielsweise durch Netzwerk **34M** bereitgestellt werden.

[0053] Indem wir zuerst den allgemeinen Betrieb der Kommunikationsmechanismen des Netzwerkes **34** eines HAN File Servers **10** betrachten und uns auf **Fig.** 1 und 2 beziehen, unterstützt jedes Blade **14** eines HAN File Servers **10** eine Vielzahl von Ports **34P**, die mit den Netzwerken **34** verbunden sind und mit diesem kommunizieren. Bei einer vorliegenden Realisierung unterstützt jedes Blade **14** beispielsweise insgesamt fünf Ports **34P**, wobei vier Ports **34P** mit den Netzwerken **34N** verbunden sind, um Clients **34C** zu bedienen, wobei ein Port zur Verwaltung des HAN File Servers **10** reserviert, und mit einem Verwaltungsnetzwerk **34M** verbunden ist. Wie veranschaulicht, sind entsprechende Ports **34P** auf jedem der Blades **14A** und **14B** mit demselben Netzwerk **34** verbunden, so daß jedes Netzwerk **34** durch einen Abgleich mit den Ports **34P** über eine Verbindung zu jedem der Blades **14A** und **14B** verfügt. Bei dem vorliegenden Beispiel sind die Ports **34P** des HAN File Servers **10** mit 10 verschiedenen IP-Adressen konfiguriert, d.h. eine Adresse für jeden Port, wobei die Ports **34P** eines jeden entsprechenden Paares von Ports **34P** der Blades **14** mit demselben Netzwerk **34** verbunden sind, so daß jedes Netzwerk **34** den HAN File Server **10** über zwei Adressen ansprechen kann, und zwar eine für jedes Blade **14A** und **14B**. Die Ports **34P**, denen jeder Client eines HAN File Servers **10** zugeordnet ist, werden innerhalb eines jeden Clients durch eine in dem Client residente ARP-Tabelle (Address Resolution Protocol) festgelegt, was auf diesem Fachgebiet herkömmlicherweise so vorgenommen, und von gewöhnlichen Fachleuten auf diesem Fachgebiet gut verstanden werden wird. Zusätzlich, und wie ebenfalls in **Fig.** 2 dargestellt, können Clients **34C** auf den HAN File Server **10** entweder über eine der direkt verbundenen Netzwerk 34-Verbindungen oder über den optionalen Router **34R** zugreifen, wenn der HAN File Server **10** mit einem Vorgabeleitweg konfiguriert, oder mit einem Leitwegprotokoll wie beispielsweise RIP (Rest In Peace/nicht behebbarer Programmabbruch) oder OSP ausgestattet ist. Bei alternativen Realisierungen eines HAN File Servers **10** kann jeder Client **34C** mit den Ports **34P** des HAN File Servers **10** über mehrere Netzwerke **34** verbunden sein, und bei den Netzwerken **34** können unterschiedliche Technologien zur Anwendung kommen, beispielsweise Local Area Networks (LANs), Wide Area Networks (WANs), direkte Prozessorverbindungen oder Busse, Faseroptikverbin-

dungen oder eine Kombination aus denselben, mit angemessenen Anpassungen der ARP-Tabellen von Clients **34C** und des HAN File Servers **10**, die weiter unten beschrieben werden.

[0054] Wie in **Fig.** 3 dargestellt, weisen die auf jedem FEP **44F** eines jeden Blade **14A** und **14B** residenten Mechanismen von Network **48** weiterhin CIFS 62- und NFS 64-Netzwerkdateisysteme und andere notwendige Dienste auf. Diese zusätzlichen Dienste, die nicht explizit in **Fig.** 3 dargestellt sind, umfassen folgende:
NETBIOS – ein Microsoft/IBM/Intel-Protokoll, das von PC-Clients verwendet wird, um auf Fernressourcen zuzugreifen. Eines der Schlüsselmerkmale dieses Protokolls besteht darin, daß es Servernamen in Teilnehmeradressen auflöst, wobei ein Server eine Komponente eines UNC-Namens ist, der von dem Client benutzt wird, um den gemeinsam genutzten Anteil zu identifizieren, d.h. a \\server\share, wobei der Server im HAN File Server **10** das eine Blade **14A** oder **14B** darstellt. NETBIOS stellt auch die CIFS 62-Paketrahmung bereit, und der HAN File Server **10** verwendet NETBIOS über TCP/IP, wie in RFC1001 und RFC1002 definiert;
SNMP – Simple Network Management Protocol, das den HAN File Server **10** mit einem Prozeß ausstattet, Agent genannt, der Informationen über das System bereitstellt und über die Fähigkeit verfügt, Auffangroutinen zu senden, wenn interessante Ereignisse eintreten;
SMTP – Simple Mail Transport Protocol, das von dem HAN File Server **10** zum Senden von Email-Mitteilungen verwendet wird, wenn interessante Ereignisse eintreten;
NFS – Sun Microsystems Network Information Service, der ein Protokoll bereitstellt, das von NFS-Servern zu Identifikation von Benutzer-IDs verwendet wird, um den Zugriff auf NFS-Dateisysteme zu kontrollieren; und,
RIP – ein dynamisches Leitwegprotokoll, das verwendet werden kann, um Netzwerktopologie zu entdecken, um Clients zu unterstützen, die hinter einem solchen Router laufen, wie beispielsweise Router **34R**. Bei der vorliegenden Realisierung eines HAN File Servers **10** arbeitet dieses Protokoll im Passivmodus, um Leitweginformationen zu überwachen. Bei alternativen Realisierungen kann der Benutzer einen vorgegebenen Leitweg während der Systeminitialisierung installieren oder bestimmen.

[0055] Zum Zwecke der Beschreibung der vorliegenden Erfindung wird es von gewöhnlichen Fachleuten auf diesem Fachgebiet verstanden werden, daß beim normalen Betrieb eines HAN File Servers **10** die Elemente von jedem Network **48**, d.h. NetSOS **56**, TCP/IP Stack **58**, NetDDs **60** und CRT **48A** auf eine herkömmliche Art und Weise arbeiten, die von gewöhnlichen Fachleuten auf diesem Fachgebiet gut verstanden wird, um Netzwerkkommunikationsvorgänge zwischen den Clients **34C** und dem HAN File Server **10** auszuführen. Als solche werden diese Aspekte des HAN File Servers **10** und eines Network **48** nicht detaillierter abgehandelt, wobei sich die nachfolgenden Abhandlungen auf die mit Hochverfügbarkeitsnetzwerken in Zusammenhang stehenden Kommunikationsmechanismen eines HAN File Servers **10** konzentrieren werden.

Kommunikationsausfallbehandlungsmechanismen des HAN File Servers **10** (**Fig.** 1, 2 und 3):

Netzwerk-Kommunikationsausfallsmechanismen:

[0056] Es wird von gewöhnlichen Fachleuten auf diesem Fachgebiet erkannt und verstanden werden, daß, während ein Kommunikations- oder Konnektivitätsausfall leicht erkannt wird, die Feststellung, welche Komponente ausgefallen ist, und daher auch die der passenden Korrekturmaßnahmen, schwierig und komplex ist. Beispielsweise umfassen mögliche Ausfallquellen einen ausgefallenen Port **34P**, eine ausgefallene Verbindung zwischen einem Port **34P** und einem Hub oder Schalter des Netzwerkes **34**, oder eine ausgefallene oder fehlerhafte Partition im Netzwerk zwischen den Blades **14**, sind aber nicht auf diese begrenzt. Ein HAN File Server **10** stellt jedoch IP-Netzwerkkommunikationsdienste bereit, die dazu in der Lage sind, Ausfälle einer oder mehrerer Netzwerk 34-Schnittstellen und verschiedener Arten von Netzwerk 34-Ausfällen zu verwalten, sowie Blade 14-Ausfälle und, um das Serversystem mit der Fähigkeit des stufenweisen Leistungsrückgangs aufgrund verschiedener Ausfälle auszustatten, realisiert eine Anzahl zusammenwirkender oder komplementärer Mechanismen, um die verschiedenen Klassen oder Arten von Ausfällen zu bewältigen. Beispielsweise bei einem Port 34P-Schnittstellenausfall bei einem Blade **14** kann der HAN File Server **10** die Rechenblattelementbus 30-Verbindung zwischen den Blades **14A** und **14B** verwenden, um den Netzwerkbetrieb von dem funktionierenden entsprechenden Port **34P** auf dem gleichrangigen Blade **14** zu dem Blade **14** weiterzuleiten, in dem der Port **34P** ausfiel. Durch diese Einrichtung wird die Notwendigkeit vermieden, daß das gesamte Blade **14** als Ergebnis des Ausfalls eines einzelnen, darin residenten Netzwerk-Ports **34P** ausfallen muß, und ohne die daraus folgende Notwendigkeit, die durch das Blade **14** unterstützten Dateisysteme zu verlegen. Es wird erkannt werden, daß diese Einrichtung auch mehrere Netzwerk-Port 34P-Ausfälle auf einem oder beiden der Blades **14** aufnimmt, d.h. solange die Ausfälle bei unterschiedlichen Netzwerken **34** eintreten, d.h. solange wie Fehler nicht auf beiden der entsprechenden Paare von Ports **34P** auf den Blades **14** eintreten. Solange noch mindestens ein Port **34P** auf einem der Blades **14** für jedes Netzwerk **34** vorhanden ist, werden die Clients keine Ausfälle erkennen.

[0057] Die Hochverfügbarkeitskommunikationsmechanismen eines HAN File Servers **10** werden durch einen in jeder Domäne der Blades **14** residenten Kommunikations-Failover-Mechanismus (CFail) **66** bereitgestellt und umfassen separat arbeitende, jedoch zusammenwirkende Mechanismen zur Kommunikationsausfallbehandlung in bezug auf die Mechanismen von Network **48** eines jeden Blade **14**, und der Message 42-Mechanismen von Blades **14A** und **14B**.

[0058] Indem wir zuerst die Funktionen und Vorgänge von CFail **66** in bezug auf Network **48** betrachten, d.h. die Kommmunikationen zwischen Clients **34C** und der Steuerungs-Prozessor-Subsystem 14-Domäne, kann ein CFail **66** einen als IP-Übergabe bezeichneten Vorgang ausführen, wodurch die einem Blade **14** zugeordneten ausgefallenen Netzwerk 34-Dienste auf die entsprechenden, nicht ausgefallenen Ports **34P** des entgegengesetzten gleichrangigen Blade **14**, und, wie weiter unten beschrieben, über alternative Pfade über die Blades **14** geleitet werden. Wie in **Fig.** 3 veranschaulicht, umfaßt jeder CFail **66** einen Kommunikationsüberwachungsprozeß-Protokollmechanismus (CMonitor) **66C**, der im FEP **44F** des Blade **14** resident ist und die Überwachung und Koordination aller Kommunikationsfunktionen der Blades **14** ausführt, einschließlich Vorgängen des NetSOS **56** der Blades **14A** und **14B**, Kommunikationen über Ports **34P** und Netzwerke **34**, und Kommunikationen über den Blattelementbus 30-Pfad zwischen den Blades **14A** und **14B**. Zum Zwecke der Überwachung und Fehlererkennung bei Kommunikationen über Ports **34P** und Netzwerke **34** umfaßt jeder CFail **66** eine SLIP-Schnittstelle (SLIP) **66S**, die über Network **48** und Ports **34P** des Blade **14** arbeitet, in dem sie resident ist, um Netzwerkkoordinationspakete/Network Coordination Packets (NCPacks) **66P** mit dem entgegengesetzten gleichrangigen Blade **14** auszutauschen. Die NCPacks **66P** enthalten beispielsweise die Netzwerkaktivitäts-Koordinationsinformationen und -benachrichtigungen, und werden von CMonitor **66C** verwendet, um ausgefallene Ports **34P** zu erkennen und zu identifizieren. Jede SLIP **66S** überträgt insbesondere regelmäßig ein Peil-NCPack **66P** an die SLIP **66S** und den CMonitor **66C** des entgegengesetzten gleichrangigen Blade **14** über jeden Netzwerk 34-Pfad zwischen den Blades **14**. Ein Netzwerk 34-Pfad zwischen den Blades **14** wird erkannt und als ausgefallen betrachtet, wenn der CMonitor **66C** eines Blade **14** während eines zuvor festgelegten Ausfallerkennungsintervalls kein Peil-NCPack **66P** von dem entgegengesetzten gleichrangigen Blade **14** über den Pfad erhält, wobei angenommen wird, daß der Ausfall in der Port 34P-Schnittstelle des entgegengesetzten Blade **14** eintrat. Das zuvor festgelegte Ausfallerkennungsintervall ist länger als das Intervall zwischen NCPack 66P-Übertragungen und ist typischerweise kürzer als das CIFS-Client-Auszeitintervall. Bei einer beispielhaften Realisierung kann dieses Intervall etwa 5 Sekunden bei einem CIFS-Auszeitintervall von 15 Sekunden betragen.

[0059] Wie in **Fig.** 3 dargestellt, weist jeder CFail **66** einen ARP-Antworterzeuger/ARP Response Generator (ARPGen) **66G** auf, der auf einen CMonitor **66C** reagiert, um unaufgeforderte ARP-Antworten **66R** und einen Pfadmanager/Path Manager (PM) **66M** zu erzeugen, der die Inhalte von in CRT **48A** residenten CREs **48E** in Übereinstimmung mit den Vorgängen von CFails **66** verwaltet, um die Umleitung von Client 34C-Kommunikationen durch Network **48** zu verwalten. Wenn der CMonitor **66C** eines Blade **14** einen Kommunikationspfadausfall in dem gleichrangigen Blade **14** feststellt, wie beispielsweise einen Ausfall in der Port 34P-Schnittstelle, werden diese Informationen an den ARPGen **66G** weitergeleitet, der eine entsprechende unaufgeforderte ARP-Antwort **66R** an die vom Port **34P** aus, dem der Ausfall zu geordnet ist, verbundenen Clients erzeugt. Eine ARP-Antwort **66R** bewirkt die Abänderung oder das erneute Schreiben von Informationen in den ARP-Tabellen der Ziel-Clients **34C**, um die Clients **34C** zu dem funktionierenden Port **34P** des Paares von entsprechenden Ports **34P** umzuleiten, d.h. der Port **34P** des CFail **66**, der die ARP-Antwort **66R** erzeugt. Genauer ausgedrückt versucht eine unaufgeforderte ARP-Antwort **66R**, die durch einen ARPGen **66G** übertragen wird, die in jedem dieser Clients **34C** residente ARP-Tabelle abzuändern oder neu zu schreiben, um Kommunikationen von diesen Clients **34C** zu dem entsprechenden Port **34P** des Blade **14** umzuleiten, das den AR-PGen **66G** umfaßt, der die ARP-Antwort **66R** überträgt. Jeder CFail **66** versucht dadurch, die Clients **34C** des ausgefallenen Kommunikationspfades auf den entsprechenden Port **34P** des Blade **14** umzuleiten, in dem der CFail **66** resident ist, was zur Folge hat, daß, wie weiter unten beschrieben werden wird, eine Umleitung der mit dem ausgefallenen Port **34P** kommunizierenden Clients zu dem noch funktionsfähigen entsprechenden Port **34P** des Blade **14** erfolgt, das den funktionsfähigen Port **34P** enthält.

[0060] Zusätzlich reagiert der PM **66P** eines jeden Blade **14** auf die Vorgänge des CMonitors **66C** und auf die Erzeugung einer oder mehrerer ARP-Antworten **66R** durch den ARPGen **66G** durch Abänderung der CREs **48E** der CRT **48A** entsprechend den Clients **34C**, die das Ziel der ARP-Antworten **66R** sind. Der PM **66P** schreibt insbesondere einen Ausfallseintrag/Failed Entry (FE) **48F** in den jedem Client **34C** entsprechenden CRE **48E**, an den eine ARP-Antwort **66R** gerichtet war, was darauf hinweist, daß die Kommunikationen des entsprechenden Clients **48C** umgeleitet wurden, und setzt ein Durchgangsfeld/Passthrough Field (PF) **48P** in die CRT **48A** ein, um jedem Network **48** mitzuteilen, daß die Blades **14** in einem Modus arbeiten.

[0061] Danach, und nach Empfang irgendwelcher Kommunikationen über seine eigenen Ports **34P** von einem Client **34C**, die an das gleichrangige Blade **14** gerichtet sind, d.h. an ein Client-Dateisystem, das auf dem gleichrangigen Blade **14** unterstützt wird, überprüft Network **48** das PF **48P** um festzustellen, ob der Durchgangs-Betriebsmodus wirksam ist. Wenn der Durchgangsmodus wirksam ist, richtet Network **48** die Kommunikation über den aus dem Blattelementbus 30-Pfad zwischen den BEPs **44B** des Blade **14** bestehenden Durchgangspfad an das gleichrangige Blade **14**. Zusätzlich, und als Ergebnis einer Umleitung, wie sie gerade beschrieben wurde, kann ein Network **48** eine Kommunikation über einen Blattelementbus **30** über den Blattelementbus 30-Durchgangspfad empfangen, der auf einen Port **34P** in seinem Blade **14** umgeleitet war, jedoch durch den Blattelementbus 30-Durchgangspfad mittels Umleitung über das andere Blade **14** umgeleitet wurde. In solchen Fällen reagieren CMonitor **66C** und PM **66M** auf den Empfang einer solchen Kommunikation durch Network **48** mit der Abänderung des dem Client **34C** entsprechenden CRE **48E**, der die Quelle der Kommunikation war, um Kommunikationen zu diesem Client **34C** über den Blattelementbus 30-Durchgangspfad und das gleichrangige Blade **14** zu leiten und dadurch die Umleitung von Kommunikationen in beiden Richtungen entlang des Pfades zu und von den betroffenen Clients **34C** abzuschließen.

[0062] Es wurde weiter oben beschrieben, daß bei alternativen Realisierungen eines HAN File Servers **10** jeder Client **34C** mit den Ports **34P** des HAN File Servers **10** über mehrere Netzwerke **34** verbunden sein kann, wobei für die Netzwerke **34** unterschiedliche Technologien verwendet werden können, beispielsweise Local Area Networks (LANs), Wide Area Networks (WANs), direkte Prozessorverbindungen oder Busse, Faseroptikverbindungen oder eine Kombination aus denselben. Bei diesen Realisierungen arbeiten die CFail 66-Mechanismen wie weiter oben beschrieben in bezug auf erkannte Ausfälle der Netzwerk 34-Kommunikationen, können jedoch zusätzlich unter den verfügbaren und funktionierenden alternativen Netzwerk 34-Pfaden zwischen einem Client **34C** und einem Blade **14** mit einem Port 34C-Ausfall auswählen, sowie Client 34C-Kommunikationen auf das noch funktionsfähige Blade **14** umleiten. Bei dieser Realisierung ändern die CFail 66-Mechanismen die Client 34C-ARP-Tabellen und die CREs **48E** wie weiter oben beschrieben ab, um die Client 34C-Kommunikationen umzuleiten, treffen jedoch bei der Auswahl eines alternativen Pfades eine Auswahl unter zusätzlichen Optionen.

[0063] Es muß in bezug auf die oben beschriebenen IP-Durchgangsvorgänge bemerkt werden, daß die CFail 66-Mechanismen eines HAN File Servers **10** nicht versuchen, die Position oder den Grund einer Verbindung zwischen den Netzwerken **34** und den Blades **14** zu identifizieren. Jeder CFail **66** nimmt stattdessen an, daß der Ausfall in der Port 34P-Schnittstelle des entgegengesetzten Blade **14** eintrat und initiiert dementsprechend einen IP-Durchgangsvorgang, so daß IP-Durchgangsvorgänge für einen vorgegebenen Kommunikationspfad durch die Blades **14A** und **14B** gleichzeitig ausgeführt werden können. Gleichzeitig durch die Blades **14A** und **14B** ausgeführte IP-Durchgangsvorgänge kollidieren jedoch bei der vorliegenden Erfindung nicht mehr miteinander. Das heißt, wenn beispielsweise die IP-Durchgangsvorgänge ein Ergebnis eines Ausfalls in einer Port 34P-Schnittstelle eines der Blades **14A** und **14B** oder in einer Netzwerk 34-Verbindung zu einem der Blades **14A** und **14B** sind, ist der CFail **66** des Blade **14**, dem der Ausfall zugeordnet ist, nicht dazu in der Lage, seine ARP-Antwort **66R** den über den Port **34P** oder die Netzwerk 34-Verbindung angeschlossenen Clients **34C** mitzuteilen. Folglich ist der CFail **66** des mit dem Ausfall in Zusammenhang stehenden Blade **14** nicht dazu in der Lage, den Betrieb des entsprechenden Client **34C** zu seinem Blade **14** umzuleiten. Der CFail **66** des entgegengesetzten Blade **14**, d.h. des nicht mit dem Ausfall in Zusammenhang stehenden Blade **14**, ist jedoch bei der Übertragung seiner ARP-Antwort **66R** an die mit dem ausgefallenen Pfad in Zusammenhang stehenden Clients **34C** erfolgreich, und somit auch bei der Umleitung des entsprechenden Betriebes von Client **34C** zu seinem Blade **14**. Bei einem Ausfall, der von einer Partition im Netzwerk herrührt, können beide Port 34P-Schnittstellen die Netzwerkpartition durch den Blattbus 30-Kommunikationspfad zwischen den Blades **14A** und **14B "**überbrücken", wie weiter unten beschrieben werden wird, so daß als Ergebnis alle Clients **34C** dazu in der Lage sind, mit einem beliebigen der beiden Blades **14A** und **14B** zu kommunizieren.

[0064] Schließlich werden bei einem vollständigen Ausfall eines der beiden Blades **14A** und **14B** IP-Durchgangsvorgänge durch CFails **66** auf die oben in bezug auf die Annahme der Dienste eines ausgefallenen Ports **34P** durch den entsprechenden, noch funktionsfähigen Port **34P** des anderen Blade **14** mit der Ausnahme ausgeführt, daß die Netzwerkdienste aller Ports **34P** des ausgefallenen Blade **14** von den entsprechenden Ports **34P** des noch funktionsfähigen Blade **14** übernommen werden. Es wird von gewöhnlichen Fachleuten auf diesem Fachgebiet verstanden werden, daß bei einem vollständigen Ausfall eines Blade **14** die TCP-Verbindungen des durch das ausgefallene Blade **14** bedienten Clients unterbrochen sind und nach Abschluß des IP-Durchgangs wiederhergestellt werden müssen, nachdem die auf dem Blade **14** verfügbaren Dienste auf dem noch funktionsfähigen Blade **14** verfügbar sind, und die Clients des ausgefallenen Blade **14** die TCP-Verbindungen wieder aufbauen können, jedoch zu dem noch funktionsfähigen Blade **14**.

[0065] Schließlich wird in bezug auf den Betrieb der oben beschriebenen IP-Durchgangsmechanismen verstanden werden, daß die mit Netzwerk **34** in Zusammenhang stehenden, durch einen HAN File Server **10** unterstützten Kommunikationsvorgänge wie erforderlich, gesendete Kommunikationen umfassen, beispielsweise durch die NetBIOS-Mechanismen von Network **48** sowie die oben abgehandelten Punkt-zu-Punkt-Kommunikationen, oder von Client **34C** zu HAN File Server **10**, umfassen. Wie von gewöhnlichen Fachleuten auf diesem Fachgebiet verstanden werden wird, unterscheiden sich gesendete Kommunikationen von Punkt-zu-Punkt-Kommunikationen dadurch, daß sie an eine Vielzahl von Empfängern, und nicht an einen spezifischen Empfänger gerichtet sind, jedoch ansonsten, wenn die Blades **14** im Durchgangsmodus arbeiten, auf eine Art und Weise verwaltet werden, die Client 34C-Kommunikationen ähnlich ist. In diesem Fall überprüft ein Network **48**, das eine gesendete Kommunikation empfängt, ob die Blades wie weiter oben beschrieben im Durchgangsmodus arbeiten, und leiten, wenn dies der Fall ist, jede dieser gesendeten Kommunikationen an Network **48** des entgegengesetzten Blade **14** über den Blattbus 30-Durchgangspfad weiter, woraufhin die Kommunikation durch das andere Network **48** auf dieselbe Art und Weise behandelt wird, wie eine direkt empfangene gesendete Kommunikation.

[0066] Schließlich ist es in bezug auf das oben Erwähnte gewöhnlichen Fachleuten auf diesem Fachgebiet bekannt und wird von ihnen gut verstanden, daß die CIFS-Spezifikation nach Industriestandard nicht die Wirkungen einer abgebrochenen Verbindung einer auf einem Client-System laufenden Anwendung beschreibt oder spezifiziert. Die Erfahrungen, Versuche und Anwendungsdokumentationen zeigen, daß die Wirkungen einer abgebrochenen TCP-Verbindung bei einer Anwendung anwendungsabhängig sind, und jede den Ausfall unterschiedlich behandelt. Beispielsweise schreiben bestimmte Anwendungen vor, daß Clients die Ausführung des Vorgangs unter Verwendung der TCP-Verbindung erneut versuchen sollten, wobei einige Anwendungen automatisch die erneute Ausführung des Vorgangs versuchen, während andere einen Ausfallbericht an den Benutzer ausgeben. Als solche integriert die gegenwärtig bevorzugte Realisierung von Netzwerk-Port-Failover-Mechanismen Funktionen zur Realisierung dieser Merkmale, einschließlich Funktionen in den NetDDs **60**, die die Ports **34P** steuern, um mehrere IP-Adressen zu unterstützen und es somit jedem Port **34P** zu ermöglichen, auf mehrere Adressen zu reagieren, und die für die Übertragung von IP-Adressen von einem ausgefallenen Blade **14** und für die Instantiierung der IP-Adressen auf dem noch funktionsfähigen Blade **14** notwendige Funktionalität. Der Netzwerk-Port-Failover-Mechanismus umfaßt auch weiter oben bereits abgehandelte Funktionen, um unaufgeforderte ARP-Antworten **66Rs** an mit den ausgefallenen Ports **34P** verbundene Clients **34C** zu erzeugen und zu übertragen, um die IP-Adressen in den ARP-Tabellen der Clients zu ändern, um auf die neuen Ports **34P** hinzuweisen, um eine Schnittstelle mit Verfügbarkeits- und Ausfallüberwachungsfunktionen in anderen Subsystemen zu bilden um zu wissen, wann ein vollständiger Ausfall des Blade **14** eintrat, und um die Auflösung des NetBIOS-Namens für den Ressourcennamen des ausgefallenen Blade **14** zu realisieren.

[0067] Es ist daher offensichtlich, daß die CFail 66-Mechanismen eines HAN File Servers **10** dazu in der Lage sind, Kommunikationen zwischen Clients **34C** und den Blades **14** des HAN File Servers **10** unabhängig von dem Netzwerkniveau, auf dem ein Ausfall eintritt, zu stützen oder wiederherzustellen, einschließlich auf dem Sub-Netzwerkniveau innerhalb der Port 34P-Schnittstellen der Blades **14A** und **14B**. Die einzige Anforderung besteht darin, daß ein funktionierender Netzwerkpfad und Netzwerkschnittstelle für jedes Netzwerk **34** auf wenigstens einem der Blades **14A** und **14B** vorhanden ist. Die CFail 66-Mechanismen der vorliegenden Erfindung vermeiden dadurch die für die Identifizierung und Isolation der Quelle und des Grundes von Netzwerkkommunikationsausfällen notwendigen komplexen Mechanismen und Verfahren, die typisch für den Stand der Technik sind, während sie auch die komplexen Mechanismen und Vorgänge vermeiden, die ebenfalls typisch für den Stand der Technik, und zur Koordination, Synchronisation und Verwaltung potentiell miteinander kollidierender Fehlerverwaltungsvorgänge erforderlich sind.

Blade **14**/Kommunikations- und Fehlerbehandlungsmechanismen von Blade **14**:

[0068] Es wurde weiter oben beschrieben, daß die mittlere Schicht der Kommunikationsmechanismen eines HAN File Servers **10** die Kommunikationsmechanismen umfaßt, die Kommunikationen zwischen und innerhalb der Domänen des Blade **14A** und **14B** der Steuerungs-Prozessor-Subsystem 14-Domäne unterstützen, wie beispielsweise Blattelementbus **30** und Messages **42**. Wie beschrieben, werden beispielsweise Blattelementbus 30-Pfad und Messages **42** für eine Bandbreite administrativer und verwaltungsmäßiger Kommunikationen des HAN File Servers **10** zwischen den Blades **14** als ein Segment des Dateitransaktionsverarbeitungspfades im Falle eines Kommunikationsübernahmevorgangs, und bei CMirror 54M- und LMirror 54L-Vorgängen verwendet.

[0069] Wie abgehandelt und in **Fig.** 2 dargestellt, besteht der Blattelementbus 30-Kommunikationspfad zwischen den Blades **14** aus Blattelementbus **30**, und in jedem Blade **14**, aus dem im BEP **44B** residenten BE BusSys **38O**, das solche Elemente umfaßt wie SER-DESs **38R**, FCCs **38Q**, P-P Bridges

38H, MCHs **38C** und Prozessoren **36A**. Obwohl in **Fig.** 2 nicht explizit dargestellt, wird verstanden werden, daß die BE BusSys **38O** auch in Prozessor **36A** ausführende BE BusSys 38O-Steuerungs- und Kommunikationsmechanismen umfassen, d.h. im BEP **44B**, die im allgemeinen auf die von gewöhnlichen Fachleuten auf diesem Fachgebiet gut verstandene Art und Weise arbeiten, um Kommunikationsvorgänge über BE BusSys **38O** und Blattelementbus **30** auszuführen. Es wird auch verstanden werden, daß die Prozessoren **36A** und **36B** des FEP **44F** und BEP **44B** eines jeden Blade **14** ebenfalls Message 42-Steuerungs- und Kommunikationsmechanismen ausführen, die in **Fig.** 2 oder 3 nicht explizit dargestellt, die im allgemeinen auf die von gewöhnlichen Fachleuten auf diesem Fachgebiet gut verstandene Art und Weise arbeiten, um Kommunikationsvorgänge über Message **42** auszuführen.

[0070] Die Messages **42**, die wiederum Kommunikationen zwischen den BEPs **44B** und FEPs **44A** bereitstellen, bestehen aus einem gemeinsam genutzten Meldungskommunikationsraum in dem Speicher **38A** eines jeden Blade **14**, und aus über in die Prozessoren **36A** und **36B** ausführenden Meldungsmechanismen, d.h. die im allgemeinen auf die von gewöhnlichen Fachleuten auf diesem Fachgebiet gut verstandene Art und Weise arbeiten, um Kommunikationsvorgänge über Messages **42** auszuführen.

[0071] Wie in **Fig.** 3 angegeben, umfaßt CFail **66** einen Fehlerbehandlungsmechanismus, der separat und unabhängig von SLIP **66S**, CMonitor **66C** und ARPGen **66G** ist, die in Verbindung mit Kommunikationen in und von der Steuerungs-Prozessor-Subsystem 14-Domäne arbeiten, um Fehlerbehandlungen in bezug auf Kommunikationen zwischen, und innerhalb der Domänen der Blades **14A** und **14B** der Steuerungs-Prozessor-Subsystem 14-Domäne auszuführen. Wie in dieser Figur dargestellt, umfaßt der Domänenkommunikationsfehlerbehandlungsmechanismus von CFail **66** zwischen den Blades **14** einen Blattelement-Kommunikationsmonitor/Blade Communications Monitor (BMonitor) **66B**, der den Betrieb der Blattelementbus 30-Kommunikationsverbindung zwischen den Blades **14A** und **14B** überwacht, worin Blattelementbus **30** und das BE BusSys **38O** des Blade **14** enthalten sind, und den Betrieb von Message **42** des Blade **14**, obwohl diese Verbindung nicht explizit in **Fig.** 3 dargestellt ist. Indem wir zuerst Blattelementbus **30** betrachten, wird bei einem Ausfall des Kommunikationspfades von Blattelementbus **30** zwischen den Blades **14** aus irgendeinem Grund, d.h. in Blattelementbus **30** oder BE BusSys **38O**, dieser Ausfall durch BMonitor **66B** erkannt, und zwar typischerweise durch Benachrichtigung von den Steuerungsmechanismen von BE BusSys **38O**, die in die Prozessoren **36A** ausführen, daß eine versuchte Kommunikation über den Blattelementbus 30-Pfad nicht als empfangen anerkannt wurde.

[0072] Bei einem Ausfall des Blattelementbusses 30-Kommunikationspfades liest BMonitor **66B** die Blattelement-Leitwegtabelle (BRT) **48B**, in der die Informationen in Bezug auf die verfügbaren Kommunikationsleitwegpfade zwischen den Blades **14A** und **14B** gespeichert sind. Die dort gespeicherten Pfadinformationen umfassen beispielsweise Leitweginformationen für Kommunikationen über den Blattelementbus **30**, aber auch Leitweginformationen für Kommunikationen für die verfügbaren Netzwerk 34-Pfade zwischen den Blades **14A** und **14B**. Es wird bemerkt werden, daß die BRT **48B** in Verbindung mit CFail **66** gespeichert sein kann, aber, wie in **Fig.** 3 dargestellt, bei den gegenwärtig bevorzugten Ausführungen der BRT **48B** der Blades **14** in Verbindung mit Network **48** resident ist, da die für die Netzwerke **34** relevanten Leitweginformationen für Network **48** bei normalem Betrieb von Network **48** leicht verfügbar und zugänglich sind, wie beispielsweise beim Aufbau einer CRT **48A**. BMonitor **66B** liest die Leitweginformationen in bezug auf die verfügbaren Kommunikationspfade zwischen den Blades **14**, mit Ausnahme des Pfades des Blattelementbusses **30** aufgrund des Ausfalls dieses Pfades, und wählt einen verfügbaren Netzwerk 34-Pfad zwischen den Networks **48** der Blades **14**, der als Ersatz- oder Austauschpfad für den Blattelementbus 30-Pfad verwendet werden soll. In dieser Hinsicht ist zu bemerken, daß BMonitor **66B** die Inhalte der BRT **48B** während aller IP-Durchgangsvorgänge auf dieselbe Art und Weise, und aktuell mit der Abänderung der CREs **48E** der CRT **48A** durch PM **66M** abändert, um nicht funktionierende Netzwerk 34-Pfade zwischen den Blades **14** anzugeben, so daß der Ersatzpfad für den Blattelementbus 30-Pfad nur aus funktionsfähigen Netzwerk 34-Pfaden ausgewählt wird.

[0073] BMonitor **66B** gibt dann eine Benachrichtigung an das BE BusSys **38O** und die Steuerungs- und Kommunikationsmechanismen von Message **42** aus, die in FEP **44F** und BEP **44B** ausführen, die alle Kommunikationen umleiten, die zu dem Blattelementbus 30-Pfad geleitet werden würden, und zwar entweder direkt durch BEP **44B** oder indirekt über Message **42** durch FEP **44F** zu Network **48** und durch den von PM **66M** ausgewählten Netzwerk 34-Pfad.

[0074] Bei einem Ausfall des Blattelementbus 30-Kommunikationspfades zwischen den Blades **14** aus irgendeinem Grund, arbeiten die Mechanismen von CMonitor **66C** und BMonitor **66B** deshalb darauf hin, einen alternativen Kommunikationspfad für Blade **14** zu Blade 14-Kommunikationen über Netzwerk **34** zu finden und zu verwenden. In dieser Hinsicht sollte erneut bemerkt werden, daß die CFail 66-Mechanismen nicht versuchen, die Position oder den Grund des Ausfalls zu identifizieren und dadurch die für die Identifizierung und Isolation der Quelle und des Grundes von Netzwerkkommunikationsausfällen typischerweise notwendigen komplexen Mechanis-

men und Verfahren, sowie die komplexen Mechanismen und Vorgänge vermeiden, die zur Koordination, Synchronisation und Verwaltung potentiell miteinander kollidierender Fehlerverwaltungsvorgänge typischerweise notwendig sind.

[0075] Es ist auch zu bemerken, daß die Kommunikationsausfallbehandlungsmechanismen eines HAN File Servers **10** separat und unabhängig voneinander arbeiten und so erneut die Verwendung von komplexen Mechanismen und Vorgängen vermeiden, die zur Koordination, Synchronisation und Verwaltung potentiell miteinander kollidierender Fehlerverwaltungsvorgänge typischerweise notwendig sind, die jedoch bei der Behandlung mehrerer Ausfallsquellen oder mehrerer Ausfälle zusammenwirkend arbeiten. So werden beispielsweise die CFail 66-Netzwerk 34-Ausfallsmechanismen, d.h. die mit CMonitor **66C** in Zusammenhang stehenden Mechanismen, unabhängig von den durch die CFail 66-Blattelementbus 30-Ausfallsmechanismen ausgeführt, d.h., die mit BMonitor **66B** in Zusammenhang stehenden Mechanismen, werden aber auf eine funktionell zusammenwirkende Art und Weise ausgeführt, um die Kommunikationen zwischen den Clients **34C** und den Blades **14** sowie zwischen den Blades **14** aufrechtzuerhalten. Die Kommunikationen werden ohne Berücksichtigung der Ausfallsquellen oder Reihenfolgen von Ausfällen aufrechterhalten, solange ein einzelner funktionsfähiger Netzwerk 34-Pfad zwischen den Blades **14** und zu jedem Client **34C** vorhanden ist, die bei einem Ausfall eines Blattelementbus 30-Pfades ausgeführt werden.

[0076] Zur Veranschaulichung: Ein Netzwerk 34-Ausfall, der mit einem ersten der Blades **14** in Zusammenhang steht, hat, wie weiter oben beschrieben, die Umleitung der Client 34C-Kommunikationen über das zweite Blade **14**, und zu dem ersten Blade **14** über die Blattelementbus 30-Verbindung zwischen den Blades **14** durch die CFail **66** Netzwerk 34-Ausfallsmechanismen zur Folge. Ein nachfolgender Ausfall der Blattbuselement 30-Verbindung hat dann das Ergebnis zur Folge, daß die Client 34-Kommunikationen, die über das zweite Blade **14** und die Blattelementbus 30-Verbindung umgeleitet wurden, erneut von dem zweiten Blade **14**, und zurück zu dem ersten Blade **14** über einen alternativen und funktionsfähigen Netzwerk 34-Pfad zwischen dem zweiten und ersten Blade **14** durch die CFail **66** Blattelementbus 30-Ausfallsmechanismen umgeleitet wird.

[0077] Ein weiteres Beispiel: Wenn der erste Ausfall in der Blattelementbus 30-Verbindung eintritt, würden die Kommunikationen zwischen den Blades **14**, wie weiter oben beschrieben, auf einen alternativen, noch funktionsfähigen Pfad zwischen den Blades **14** über Netzwerke **34** durch die CFail **66** Blattelementbus 30-Ausfallsmechanismen umgeleitet werden. Bei Eintritt eines nachfolgenden Ausfalls in diesem Netzwerk 34-Pfad würde dieser Ausfall als mit Netzwerk **34** in Zusammenhang stehender Ausfall erkannt werden, und die CFail **66** Netzwerk 34-Ausfallsmechanismen der Blades **14** würden zuerst versuchen, die zuvor zwischen den Blades **14** umgeleiteten Kommunikationen über die Bus-Blattelement 30-Verbindung umzuleiten. Die CFail **66** Blattelementbus 30-Ausfallsmechanismen würden jedoch die zuvor umgeleiteten Kommunikationen über einen verfügbaren und noch funktionsfähigen, alternativen Netzwerk 34-Pfad zwischen den Blades **14** umleiten, da die Blattelementbus 30-Verbindung nicht betriebsfähig ist.

[0078] Es ist daher offensichtlich, daß verschiedene Kombinationen und Reihenfolgen der durch die CFail **66** Netzwerk 34- und die Blattelementbus 30-Ausfallsmechanismen separat und unabhängig ausgeführten Vorgänge für jede beliebige Kombination oder Reihenfolge von Netzwerk 34- und Blattelementbus 30-Ausfällen ausgeführt werden können, um Kommunikationen zwischen Clients **34C** und den Blades **14** und den Blades **14** aufrechtzuerhalten. Erneut ist zu betonen, daß die Kommunikationen ohne Berücksichtigung der Ausfallsquellen oder Reihenfolgen von Ausfällen aufrechterhalten werden, solange ein einzelner funktionsfähiger Netzwerk 34-Pfad zwischen den Blades **14** und zu jedem Client **34C** vorhanden ist, der bei einem Ausfall eines Blattelementbus 30-Pfades ausgeführt wird.

[0079] Schließlich muß als letzter Punkt in dieser Hinsicht bemerkt werden, daß ein Ausfall in der Message 42-Verbindung zwischen dem FEP **44F** und BEP **44B** eines Blade **14** eintreten kann. In vielen Fällen ist dies die Folge eines Ausfalls, der wiederum den Ausfall des gesamten Blade **14** zur Folge hat, wobei jedoch in einigen Fällen dieser Ausfall auf die Message 42-Mechanismen begrenzt sein kann. Bei einem auf den Message 42-Mechanismen begrenzten Ausfall ist der FEP **44F** des Blade **14**, in dem der Ausfall eintrat, nicht dazu in der Lage, mit dem BEP **44B** des Blade **14** oder mit dem entgegengesetzten Blade **14** zu kommunizieren, und der BEP **44B** ist nicht dazu in der Lage, mit dem FEP **44B** des Blade zu kommunizieren, ist jedoch dazu in der Lage, mit dem BEP **44B** und FEP **44F** des entgegengesetzten Blade **14** über die Blattbuselement 30-Verbindung zwischen den Blades **14** zu kommunizieren.

[0080] Bei einer weiteren Realisierung der vorliegenden Erfindung erkennt aus diesem Grund BMonitor **66B** des Blade **14**, in dem der Message 42-Ausfall eintrat, einen offensichtlichen Ausfall des Blattelementbusses **30** in bezug auf den FEP **44F**, erkennt aber keinen Ausfall des Blattelementbusses **30** in bezug auf den BEP **44B**. Die BMonitor 66B- und CMonitor 66C-Mechanismen dieses Blade **14** können dadurch alle Kommunikationen von dem FEP **44F** zu dem BEP **44B** oder zu dem entgegengesetzten Bla-

de **14** über einen durch PM **66** ausgewählten Netzwerk 34-Pfad umleiten, und leiten alle Kommunikationen vom BEP 44B zum FEP **44F** zu einem Leitweg über Blattelementbus **30** und den für FEP **44F** ausgewählten Netzwerk 34-Pfad um, leiten aber BEP 44B-Kommunikationen nicht über Blattelementbus **30** um.

[0081] In dem Blade **14**, in dem der Ausfall nicht eintrat, erkennen die BMonitor 66B-Mechanismen einen offensichtlichen Blattelementbus 30-Pfadausfall in bezug auf die Kommunikationen zum FEP **44P** des Blade **14**, in dem der Message 42-Ausfall eintrat, aber erkennen keinen Blattelementbus 30-Pfadausfall in bezug auf Kommunikationen zum BEP **44B** dieses Blade **14**. Die BMonitor 66B-Mechanismen und CMonitor 66C-Mechanismen des Blade **14** leiten dadurch alle an den FEP **44F** des entgegengesetzten Blade **14** gerichteten Kommunikationen über einen alternativen Netzwerk 34-Pfad auf die beschriebene Art und Weise um, leiten an den BEP **44B** des entgegengesetzten Blade **14** gerichtete Kommunikationen aber nicht um.

Speicher-Subsystem 12-/Blade 14-Fehlerbehandlungsmechanismen:

[0082] Wie weiter oben beschrieben, umfaßt das unterste Niveau der Fehlerbehandlungsmechanismen eines HAN File Servers **10** die Kommunikationspfadstrukturen des Speicher-Subsystems **12** und der durch RAID **46** realisierten RAIDF 46F-Mechanismen. RAID-Dateifunktionen sind gewöhnlichen Fachleuten auf diesem Fachgebiet bekannt, werden von diesen gut verstanden, und als solche nur in dem Umfang abgehandelt, in dem dies für ein Verständnis der vorliegenden Erfindung notwendig ist. Die nachfolgenden Abhandlungen werden sich dementsprechend in erster Linie auf die Kommunikationspfadstrukturen innerhalb des Speicher-Subsystems **12** und zwischen Subsystem **12** und den Blades **14** konzentrieren.

[0083] Wie in **Fig.** 1 dargestellt und auch weiter oben beschrieben, umfaßt das Speicher-Subsystem **12** eine aus einer Vielzahl von Festplattenlaufwerken **18** bestehende Laufwerkbank **16** auf, wobei auf jedes von ihnen bidirektionale Lese-Schreibzugriffe über Doppelspeicher-Schleifenmodule **20A** und **20B** erfolgen. Die Speicher-Schleifenmodule **20A** und **20B** weisen jeweils MUXBANKs **22A** und **22B** auf, von denen jede eine Vielzahl von MUXs **24** und Schleifencontroller **26A** und **26B** aufweist, wobei die MUXs **24** und Schleifencontroller **26A** und **26B** eines jeden Schleifencontrollermoduls **20** bidirektional durch MUX-Schleifenbusse **28A** und **28B** miteinander verbunden sind. Wie dargestellt, weisen die MUXBANKs **22A** und **22B** jeweils einen MUX **24D** auf, der mit einem entsprechenden unter den Plattenlaufwerken **18** verbunden ist und diesem entspricht, so daß jedes Plattenlaufwerk **18** der Laufwerkbank **16** eine bidirektionale Lese-Schreibverbindung zu einem entsprechenden MUX **24D** in jeder der MUXBANKs **20A** und **20B** aufweist. Jede der MUXBANKs **20A** und **20B** ist weiterhin bidirektional mit dem entsprechenden der Rechenblattelemente **14A** und **14B** jeweils durch MUX **24CA** und MUX **24CB** verbunden, wobei die Rechenblattelemente **14A** und **14B** bidirektional durch den Blattelementbus **30** verbunden sind.

[0084] Daher ist jedes der Plattenlaufwerke **18** bidirektional mit einem MUX **24D** der MUX Bank **22A** und mit einem MUX **24D** der MUX Bank **22B** verbunden, und die MUXs **24** von MUX Bank **20A** sind über einen Schleifenbus **26A** miteinander verbunden, während die MUXs **24** von MUX Bank **22B** über einen Schleifenbus **26B** miteinander verbunden sind, so daß jedes Plattenlaufwerk **18** sowohl über Schleifenbus **26A** als auch über Schleifenbus **26B** zugänglich ist. Zusätzlich kommuniziert das Prozessorblatt **14A** bidirektional mit Schleifenbus **26A**, während Prozessorblatt **14B** bidirektional mit Schleifenbus **26B** kommuniziert und die Blattelementprozessoren **14A** und **14B** direkt miteinander verbunden sind und über den Blattelementschleifen (Blade) -Bus **30** kommunizieren.

[0085] Es wird daher erkannt werden, daß es sich bei den untergeordneten Kommunikationsfehlerbehandlungsmechanismen innerhalb des Speicher-Subsystems **12** im wesentlichen um eine passive Pfadstruktur handelt, die mehrfache redundante Zugriffspfade zwischen jedem Plattenlaufwerk **18** und Blattelementprozessoren **14A** und **14B** bereitstellt. Als solche können die Blattelementprozessoren **14A** und **14B** bidirektional mit einem beliebigen der Plattenlaufwerke **18** kommunizieren, entweder direkt über den ihnen zugeordneten Schleifenbus **26** oder indirekt über das andere der Blattelementprozessoren **14**, und können direkt miteinander kommunizieren, wenn ein Ausfall in einem oder mehreren Kommunikationspfaden innerhalb des Speicher-Subsystems **12** eingetreten sein sollte. Die Fehlerbehandlungsmechanismen zur Behandlung von innerhalb einem oder mehreren Plattenlaufwerken **18** eintretenden Fehlern bestehen wiederum aus den in diesem Dokument oben abgehandelten RAIDF 48F-Mechanismen.

[0086] Es wird auch erkannt werden, daß die passive Pfadstruktur des Speicher-Subsytems **12** separat und unabhängig von den Kommunikationsmechanismen und den CFail 66-Netzwerken **34** und den Blattelementbus 30-Ausfallmechanismen der Blades **14**, jedoch mit den Mechanismen der Blades **14** zusammenwirkend arbeitet, um Kommunikationen zwischen den Clients **34C** und den Plattenlaufwerken **18** sicherzustellen, in denen die Dateisysteme der Clients **34C** resident sind. Auch hier bieten diese Mechanismen ein hohes Niveau an Dateisystemverfüg-

barkeit, während sie die Verwendung von komplexen Fehlererkennungs-, Identifikations- und Isolationsmechanismen sowie die Verwendung komplexer Fehlerverwaltungskoordinations-, -synchronisations-, und Verwaltungsmechanismen vermeiden.

Dateitransaktions-Fehlerbehandlungsmechanismen eines HAN File Servers **10** und Zusammenarbeit mit den Kommunikationsausfallbehandlungsmechanismen eines HAN File Servers **10** (**Fig.** 1, 2 und 3):

[0087] Es wurde in diesem Dokument weiter oben beschrieben, daß die gegenwärtig bevorzugte Ausführung eines HAN File Servers **10** eine Anzahl von Hochverfügbarkeitsmechanismen umfaßt, d.h. Mechanismen, die es dem HAN File Server **10** ermöglichen, Clients bei einem Ausfall einer oder mehrerer Komponenten des HAN File Servers **10** ununterbrochene Dateiserverdienste bereitzustellen. Viele dieser Mechanismen sind typisch für die gegenwärtig beim Stand der Technik verwendeten, wie beispielsweise die grundlegenden RAIDF 46F-Funktionen, wie von gewöhnlichen Fachleuten auf diesem Fachgebiet gut verstanden werden wird, und werden daher in diesem Dokument nicht detailliert abgehandelt, außer in dem Umfang, in dem dies für ein vollständiges Verständnis der vorliegenden Erfindung notwendig ist.

[0088] Im allgemeinen übernehmen jedoch bei einem Ausfall einer Komponente eines HAN File Servers **10** die noch funktionsfähigen Komponenten im HAN File Server **10** durch den Betrieb der Hochverfügbarkeitsmechanismen die von der ausgefallenen Komponente ausgeführten Aufgaben und Dienste und fahren mit der Bereitstellung dieser Dienste fort. Es wird von gewöhnlichen Fachleuten auf diesem Fachgebiet geschätzt und verstanden werden, daß es bei dem Betrieb solcher Hochverfügbarkeitsmechanismen eine Anzahl von Aspekten gibt, und es erforderlich ist, daß diese Mechanismen mehrere Vorgänge ausführen, um diese Funktionen zu erfüllen. Beispielsweise müssen die Hochverfügbarkeitsmechanismen erkennen, daß eine Komponente ausgefallen ist, um die Ressourcen oder Funktionen von den ausgefallenen Komponenten zu den noch funktionsfähigen Komponenten zu übertragen oder zu bewegen, um den Status der in die noch funktionsfähigen Komponenten übernommenen Ressourcen wiederherzustellen, so daß die von den ausgefallenen Komponenten bereitgestellten Dienste und Funktionen nicht sichtbar unterbrochen sind, um den Ersatz oder die Korrektur der ausgefallenen Komponente zu ermöglichen und die Ressourcen nach der Reparatur der ausgefallenen Komponente in diese zurück zu übertragen oder zu bewegen.

[0089] Wie weiter oben in bezug auf die Kommunikationen beschrieben wurde, arbeiten die Dateitransaktions- und Kommunikationsmechanismen eines HAN File Servers **10** einzeln, und wie noch detaillierter in nachfolgenden Abhandlungen beschrieben werden wird, arbeiten die Hochverfügbarkeitsmechanismen eines HAN File Servers **10** der vorliegenden Erfindung mit einer Anzahl verschiedener Funktionsniveaus des HAN File Servers **10**. Im allgemeinen wird auf jedem Funktionsniveau eines HAN File Servers **10** eine unterschiedliche Gruppe oder Art von Vorgängen und Funktionen ausgeführt, wobei sich die Hochverfügbarkeitsmechanismen dementsprechend voneinander unterscheiden und unabhängig, jedoch zusammenwirkend arbeiten, um ein hohes Niveau an Serververfügbarkeit auf jedem Niveau und für den HAN File Server **10** als ein System bereitzustellen. Nachfolgend wird die Struktur und der Betrieb dieser Mechanismen sowie die Zusammenarbeit dieser Mechanismen noch detaillierter abgehandelt.

[0090] Das höchste Niveau an Funktionalität in einem HAN File Server **10** ist beispielsweise das Kommunikationsniveau, das Client-Kommunikationsaufgaben und -dienste ausführt, d.h. Kommunikationen zwischen den Clients und den von dem HAN File Server **10** unterstützten Client-Dateisystemen durch Netzwerke **34**. Die Kernfunktionen dieses Kommunikationsniveaus werden durch die Mechanismen von Network **48** und den damit in Zusammenhang stehenden Komponenten des HAN File Servers **10** bereitgestellt, wobei die Hochverfügbarkeitsmechanismen auf dem Kommunikationsniveau Fehlererkennungsmechanismen wie beispielsweise CFail **66** umfassen und eine Anzahl unterschiedlicher Mechanismen zur Behandlung eines Kommunikationsniveauausfalls bereitstellen. Beispielsweise bei einem Ausfall der Kommunikation über einen oder mehrere Ports **34P** eines der Blades **14A** und **14B** erkennt der CFail **66** auf dem gleichrangigen Blade **14** den Ausfall und leitet in Verbindung mit Network **48** alle Kommunikationen zwischen Clients und den ausgefallenen Ports **34P** zu den entsprechenden, noch funktionsfähigen Ports **34P** des gleichrangigen Blade **14** um. Im gleichrangigen Blade **14** leitet das darin vorhandene Network **48** die Kommunikationen auf das JFile **50** des Blade **14** mit dem ausgefallenen Port **34P** über Blattelementbus **30** zurück, so daß die ausgefallenen Ports **34P** durch die Ports **34P** des gleichrangigen Blade **14**, und der aus Blattelementbus **30** bestehende Kommunikationspfad zwischen den Blades **14** und der FEP 44F-BEP 44P-Kommunikationspfad über Message **42** umgangen wird. In dieser Hinsicht, und wie in der nächstfolgenden Abhandlung der übergeordneten Transaktionsmechanismen eines Blade **14** beschrieben werden wird, arbeiten die Hochverfügbarkeitsmechanismen von Network **48** mit denjenigen der übergeordneten Dateitransaktionsmechanismen zusammen, um offensichtlich mit Netzwerk **34** in Zusammenhang stehende Kommunikationsausfälle zu behandeln, die in der Tat und beispielsweise das Ergebnis eines Ausfalls des JFile **50** eines Blade **14** oder des gesamten Blade **14** sind.

[0091] Das nächste Funktionalitätsniveau in einem Blade **14** umfaßt die übergeordneten Dateitransaktionsfunktionen und -dienste, wobei die Kernfunktionen und -vorgänge der übergeordneten Transaktionsfunktionen durch JFile **50** und den damit in Zusammenhang stehenden übergeordneten Dateimechanismus bereitgestellt werden. Wie weiter oben beschrieben, umfassen die Hochverfügbarkeitsmechanismen des übergeordneten Dateifunktionsniveaus des HAN File Servers **10** den WCache **50C** mit CMirror **54C** und das Log **50L** mit LMirror **54L**, wobei diese Mechanismen so arbeiten, daß sie Ausfälle der übergeordneten Dateimechanismen innerhalb eines Blade **14** behandeln. Wie beschrieben, arbeitet WCache **50C** auf herkömmliche Art und Weise, um Datentransaktionen zu puffern, und CMirror **54M** ermöglicht die Wiederherstellung der Inhalte von WCache **50C** bei einem Ausfall in FEP **44F**, der sich auf WCache **50C** auswirkt. Log **50L** arbeitet wiederum mit einen Blade **14**, um eine Aufzeichnung von durch ein JFile **50** ausgeführten Dateitransaktionen zu erhalten. Hierdurch ermöglicht Log **50L** die erneute Ausführung und Wiederherstellung verlorener Dateitransaktionen, wenn beispielsweise ein Ausfall in einem JFile **50** oder Speicher-Subsystem **12** eintrat, der einen Verlust von Dateitransaktionen zur Folge hatte, bevor die Transaktionen vollständig zur festen Speicherung in dem Speicherungs-Subsystem **12** festgeschrieben wurden.

[0092] Die LMirror 54L-Mechanismen arbeiten jedoch nicht innerhalb des Blade **14**, in dem die Logs **50L** bzw. die LMirrors **54L** resident sind, doch arbeiten stattdessen Blades 14-übergreifend, so daß jeder LMirror **54L** die Inhalte des Logs **50L** des entgegengesetzten gleichrangigen Blade **14** spiegelt und erhält. Als Ergebnis halten die LMirror 54L-Mechanismen die Inhalte des entgegengesetzten gleichrangigen Blade **14** aufrecht und ermöglichen die erneute Ausführung und Wiederherstellung verlorener Dateitransaktionen in dem ausgefallenen Blade **14**, wenn das ausgefallene Blade **14** wiederhergestellt wurde, so daß es wieder in Betrieb gehen kann.

[0093] Zusätzlich sollte auch bemerkt werden, daß die LMirror 54L-Mechanismen durch Bereitstellung einer residenten Aufzeichnung möglicherweise verlorener Dateitransaktionen eines ausgefallenen Blade **14** innerhalb des noch funktionsfähigen Blade **14** es ebenso einem noch funktionsfähigen Blade **14** ermöglichen, die Unterstützung der Clients zu übernehmen, die durch ein ausgefallenes Blade **14** unterstützt wurden. Das heißt, Network **48** und JFile **50** des noch funktionsfähigen Blade **14** übernehmen die Dienste für die zuvor durch das ausgefallene Blade **14** unterstützten Clients durch Umleitung der Clients des ausgefallenen Blade **14** auf das noch funktionsfähige Blade **14**, wie weiter oben in bezug auf die Network 48-Mechanismen beschrieben. Bei diesem Prozeß, und wie weiter oben beschrieben, arbeiten die Network 48-Mechanismen des noch funktionsfähigen Blade **14** so, daß sie die IP-Adressen des ausgefallenen Blade **14** durch Umleitung der an die übernommenen IP-Adressen des JFile **50** des noch funktionsfähigen Blade **14** gerichteten Datentransaktionen übernehmen. Das JFile **50** des noch funktionsfähigen Blade **14** übernimmt die Clients des ausgefallenen Blade **14** als neue Clients unter der Annahme, daß das noch funktionsfähige Blade **14** über lokale Dateisysteme verfügt, und bedient danach diese übernommenen Clients als seine eigenen Clients, einschließlich der Aufzeichnung aller übernommenen Datentransaktionen parallel zur Behandlung der übernommenen Datentransaktionen. Das noch funktionsfähige Blade **14** verwendet sein lokales Wiederherstellungsprotokoll, d.h. den in dem noch funktionsfähigen Blade **14** residenten LMirror **54L**, um verlorengegangene Dateitransaktionen des ausgefallenen Blade **14** erneut auszuführen und zu rekonstruieren, um die Dateisysteme der Clients des ausgefallenen Blade **14** wieder in ihren erwarteten Zustand zu versetzen. In dieser Hinsicht kann das JFile **50** des noch funktionsfähigen Blade **14** festlegen, daß die "neuen" Clients von dem ausgefallenen Blade **14** übertragene Clients sind, und zwar entweder durch Benachrichtigung von Network **48** auf der Grundlage der ursprünglichen Adresse der Dateitransaktionen, wie sie an das ausgefallene Blade **14** gerichtet waren, oder durch Überprüfung der Inhalte des residenten LMirror **54L** um festzulegen, ob "neue" Client-Dateitransaktionen mit darin gespeicherten Dateitransaktionen in Wechselbeziehung stehen.

[0094] Schließlich umfaßt das niedrigste Niveau der Dateitransaktionsfunktionalität eines HAN File Servers **10** die durch RAID **46** unterstützten RAID 46-Dateitransaktionsfunktionen und -dienste. Es wird erkannt werden, daß die RAIDF 46F-Funktionen in sich selbst unabhängig von den übergeordneten Hochverfügbarkeitsmechanismen arbeiten. Es wird jedoch auch erkannt werden, daß die Kommunikationsniveau- und übergeordneten Dateitransaktionsmechanismen in Verbindung mit der Bereitstellung von alternativen Kommunikationspfaden über beispielsweise doppelte Blades **14A** und **14B**, Schleifenbusse **26A** und **26B**, und MUX-Schleifenbusse **28A** und **28B**, mit den RAIDF 46F-Funktionen zusammenwirkend arbeiten, um die Zugreifbarkeit auf die Plattenlaufwerke **18** zu erweitern.

[0095] Es ist daher aus den obigen Beschreibungen ersichtlich, daß das Kommunikationsniveau sowie die übergeordneten Dateitransaktionsmechanismen und alternativen Kommunikationspfade, die in einem HAN File Server **10** vorgesehen sind, dadurch mit den RAIDF 46F-Funktionen zusammenwirken, um die Verfügbarkeit gemeinsam genutzter Dateisystemanteile, d.h. Speicherplatz für Netzwerk-Clients, zu erweitern. Es ist auch ersichtlich, daß das in einem HAN File Server **10** bereitgestellte Kommunikations-

niveau und übergeordnete Dateitransaktionsmechanismen und alternativen Kommunikationspfade diese Ergebnisse erzielen, während sie die Verwendung von komplexen Fehlererkennungs-, Identifikations- und Isolationsmechanismen sowie die Verwendung komplexer Fehlerverwaltungskoordinations-, -synchronisations-, und Verwaltungsmechanismen vermeiden.

[0096] Zusammenfassend ist daher zu sagen, daß es aus den obigen Abhandlung ersichtlich ist, daß eine Anzahl unterschiedlicher Mechanismen zur Identifikation ausgefallener Komponenten verwendet wird, wobei der spezifische Mechanismus von der Komponente abhängig ist, d.h. von dem Subsystem des HAN File Servers **10**, in dem er resident ist sowie von den Wirkungen auf den Betrieb des HAN File Servers **10** bei Ausfall der Komponente. Beispielsweise überwachen und erkennen die RAIDM 46M-Funktionen Ausfälle bei Komponenten wie Kühlgebläsen, Stromversorgung und ähnlichen Komponenten der Blades **14A** und **14B**, während die RAIDF 46F-Funktionen Fehler und Ausfälle bei Dateisystemvorgängen der Plattenlaufwerke **18** erkennen und korrigieren und ausgleichen. Es wird erkannt werden, daß ein Ausfall bei vielen von den RAID 46-Mechanismen überwachten Komponenten die Verfügbarkeit der Daten auf dem HAN File Server 10-Niveau als einem System nicht beeinträchtigt, aber durch die Verwaltungsschnittstelle erkannt und berichtet werden muß, so daß eine Aktion zur Reparatur der Komponente eingeleitet werden kann. In einem weiteren Beispiel überwachen die Netzwerkverwaltungsfunktionen eines HAN File Servers **10** den Status der Netzwerke **34** und von mit der Netzwerk 34-Kommunikation in Zusammenhang stehenden Komponenten des HAN File Servers **10** und reagieren auf Ausfälle bei Kommunikationen zwischen dem HAN File Server **10** und den Clients des HAN File Servers **10** auf den spezifischen Ausfällen gegenüber angemessene Art und Weise. Zur Überwachung des Netzwerkes erzeugen die Netzwerkverwaltungsfunktionen Selbstprüfungen, um die eigenen Netzwerkkommunikationen des HAN File Servers **10** zu überprüfen um festzustellen, ob dieses mit dem externen Netzwerk kommuniziert. Wenn diese Selbstprüfung beispielsweise auf irgendeinem Netzwerkpfad versagt, dann werden die durch die ausgefallenen Netzwerkpfade unterstützten Kommunikationen im Failover-Verfahren auf einen anderen Netzwerkpfad übertragen, wie weiter oben beschrieben. In einem anderen Beispiel wird dieser Ausfall den Dateisystemfunktionen wie weiter oben beschrieben mitgeteilt, wenn die RAID 46-Funktionen den Ausfall eines Blade **14** erkennen, so daß die Failover-Verfahren auf dem Niveau des Dateisystems als angemessenem Niveau weiter ablaufen können.

[0097] Der nächste Schritt bei dem Ausfallbehandlungsverfahren, d.h. die Bewegung der ausgefallenen Ressourcen zu noch funktionsfähigen Ressourcen, wird typischerweise durch Neuzuweisung der Ressource an eine als noch funktionsfähig bekannte Dateilage ausgeführt. Bei einem Ausfall einer Netzwerkfunktion erfolgt die Übertragung zu einem zuvor identifizierten Netzwerkadapter, der zur Übernahme der Funktionen der ausgefallenen Vorrichtung, ebenfalls wie weiter oben beschrieben, in der Lage ist, wobei bei einem ausgefallenen Blade **14** das gleichrangige Blade **14** die Dateisysteme von dem ausgefallenen Blade **14** übernimmt.

[0098] Die Übertragung von Ressourcen von einer ausgefallenen Komponente auf eine noch funktionsfähige Komponente kann eine Umänderung des, oder Abänderung auf den Betriebszustand der Ressource erfordern, bevor die Ressource auf der noch funktionsfähigen Komponente verfügbar gemacht werden kann. Bei einer ausgefallenen Netzwerkkomponente muß beispielsweise einem bereits vorhandenen Adapter eine neue Netzwerkadresse hinzugefügt werden, und bei einem das Dateisystem betreffenden Ausfall, wie beispielsweise einem Ausfall des Blade **14**, wird das Transaktionsprotokoll erneut aufgespielt, um möglicherweise bei dem Ausfall verlorengegangene Daten zu ersetzen.

[0099] Wie zuvor beschrieben, sind viele der Komponenten des HAN File Servers **10** Hot-Swap-fähig, d.h. daß sie von dem HAN File Server **10** entfernt, und durch eine funktionierende Komponente ersetzt werden können. Sobald die Komponente ersetzt wurde, müssen die von den noch funktionsfähigen Komponenten übernommenen Ressourcen zu der ursprünglichen Komponente zurückgeführt werden, d.h. zu dem Ersatz für die ursprüngliche Komponente. Wiederherstellungsmechanismen, wie weiter oben beschrieben, bewegen dementsprechend die Ressourcen, die auf die noch funktionsfähige Komponente übertragen wurden, zu der Ersatzkomponente zurück, wobei es sich hierbei um einen Prozeß handelt, der typischerweise manuell durch den Systemverwalter zu einem Zeitpunkt initiiert wird, wenn die Unterbrechung des Dienstes annehmbar und beherrschbar ist.

Detaillierte Beschreibung der vorliegenden Erfindung (**Fig.** 4):

[0100] Nachdem nun die Struktur und der Betrieb eines HAN File Servers **10** beschrieben wurde, bei dem die vorliegende Erfindung realisiert werden kann sowie bestimmte Aspekte der vorliegenden Erfindung, wie sie beispielsweise in einem HAN File Server **10** realisiert sind, wird sich das Nachfolgende auf die vorliegende Erfindung konzentrieren und diese noch detaillierter beschreiben. Unter Bezugnahme auf **Fig.** 4 ist dort ein detailliertes Blockdiagramm der in einem Dateiserversystem **70** realisierten vorliegenden Erfindung dargestellt, wobei das Dateiserversys-

tem **70** beispielsweise in einem HAN File Server **10** realisiert ist. Aus einer Betrachtung von **Fig.** 4 wird erkannt werden, daß das Dateiserversystem **70** auf einem HAN File Server **10** basiert, und daß **Fig.** 4, die eine Realisierung des Dateiserversystems **70** veranschaulicht, beispielsweise auf den oben beschriebenen **Fig.** 1, 2 und 3 basiert.

[0101] Die Wechselbeziehung und die Beziehungen zwischen den Elementen und Vorgängen des Dateisystems **70** und einem HAN File Server **10** werden in der nachfolgenden Beschreibung der vorliegenden Erfindung abgehandelt.

[0102] Wie weiter oben in diesem Dokument beschrieben, betrifft die vorliegende Erfindung einen übergeordneten gespiegelten Transaktionsprotokollmechanismus, der in einem Doppelprozessor-Dateiserver realisiert ist, um eine fehlertolerante Datentransaktionswiederherstellung mit niedriger Latenzzeit bereitzustellen. Wie in **Fig.** 4 dargestellt, kann der übergeordnete gespiegelte Transaktionsprotokollmechanismus der vorliegenden Erfindung in einem Dateiserversystem **70** realisiert werden, das doppelte gleichrangige Dateiserver **72A** und **72B** umfaßt, die durch Blades **14A** und **14** eines HAN File Servers **10** beispielhaft dargestellt sind, oder in einem System, in dem nur ein einzelner Dateiserver **72** verwendet wird, wobei die Dateiserver **72A** und **72B** Dateiserverdienste an entsprechende Gruppen von Clients **74C** bereitstellen, und zwar beispielsweise über Netzwerke **34**. Wie weiter oben in diesem Dokument in bezug auf die Blades **14A** und **14B** eines HAN File Servers **10** beschrieben, unterstützt jeder der Dateiserver **72A** und **72B** bei normalem Betrieb eine separate und eigenständige Gruppe von Clients **74C** und exportiert oder unterstützt ein eigenständiges Set von Client-Dateisystemen (CFiles) **74F** für jede Gruppe von Clients **74C**. Das heißt, daß bei der gegenwärtig bevorzugten Ausführung des Dateiserversystems **70** keine von den Dateiservern **72A** und **72B** gemeinsam genutzten CFiles **74F** vorhanden sind.

[0103] Wie in **Fig.** 4 dargestellt, sind Dateiserverprozessoren **72A** und **72B** mit separaten Speicherplätzen ausgestattet, die durch Speicher **76A** und **76B** dargestellt, und durch die Speicher **38D** der Blades **14A** und **14B** beispielhaft dargestellt sind. Bei der gegenwärtig bevorzugten Realisierung nutzen die Dateiserverprozessoren **72A** und **72B** einen Festspeicher **78** gemeinsam, wie durch Speicher-Subsystem **12** beispielhaft dargestellt, der in der RAID-Technologie realisiert sein kann. Zum Zwecke der vorliegenden Erfindung können die unteren Niveaus des HAN-Dateisystems **10** einschließlich des internen Schreib-Pufferspeichers/Internal Write Cache (WCache) **50C** und den Dateisystemmechanismen von RAID **46**, die in dem Back-End-Prozessor (BEP) **44B** resident sind und auf diesen ausführen, funktional als Komponenten von Festspeicher **78** betrachtet werden.

[0104] Wie ebenfalls dargestellt, umfaßt jeder Dateiserver **72** einen Dateisystemprozessor (FSP) **80**, der als FSPs **80A** und **80B** dargestellt ist und die von Clients **74C** angeforderten Dateisystemtransaktionsvorgänge ausführt, und einen Kommunikationsprozessor (CP) **82**, der als CPs **82A** und **82B** dargestellt ist, und eine Hochgeschwindigkeits-Kommunikationsverbindung (CLink) **84** zwischen den Dateiservern **72A** und **72B**, und insbesondere in bezug auf die vorliegende Erfindung, zwischen den Speichern **76A** und **76B** unterstützt. Bei der weiter oben in diesem Dokument als ein HAN File Server **10** beschriebenen beispielhaften Realisierung kann jeder FSP **80** als funktional aus den übergeordneten Dateisystemfunktionen bestehend betrachtet werden, die durch das in den Front-End-Prozessor (FEP) **44F** eines Blade **14** ausführende und dort residente JFile **50** bereitgestellt werden. Wie weiter oben erwähnt, können WCache **50C** und die Dateisystemmechanismen von RAID **46**, die auf dem Back-End-Prozessor (BEP) **44B** des Blade **14** resident sind und in diesen ausführen, zum Zwecke dieser Erfindung funktional als eine Komponente des Festspeichers **78** betrachtet werden. CP **82** und CLink **84** können wiederum jeweils aus den Back-End-Bus-Subsystemen (BE BusSys's) **38O** bestehen, die auf den BEPs **44B** der Blades **14A** und **14B** und Rechenblattelementschleifenbus **30**, der die Blades **14A** und **14B** untereinander verbindet, resident sind und auf diesen betrieben werden.

[0105] Wie zuvor beschrieben, handelt es sich bei JFile **50** um ein Journaled File System, das Anforderungen **86** von Clients **74C** nach Dateisystemtransaktionen empfängt und verarbeitet, wobei es die Anforderungen **86** in entsprechende Dateisystemvorgänge/File System Operations (FSOps) **88** umwandelt. Die FSOps **88** werden dann durch einen Festschreibungsmechanismus/Commit Mechanism (Commit) **90**, der durch Commits **90A** und **90B** dargestellt ist, unter Verwendung herkömmlicher verzögerter Festschreibungsverfahren und -prozeduren als Dateisystemänderungen zur Speicherung auf dem Festspeicher **78** festgeschrieben, wobei diese von gewöhnlichen Fachleuten auf diesem Fachgebiet gut verstanden werden und typischerweise einen WCache **50C** und RAID **46** umfassen.

[0106] In dieser Hinsicht wird in einem herkömmlichen Dateiserver nach dem Stand der Technik eine Anforderung **86** von einem Client **74C** typischerweise dem Client **74C** gegenüber als abgeschlossen anerkannt, wenn der FSP **80** die Anforderung **86** angenommen, oder wenn der FSP **80** die Anforderung **86** in entsprechende FSOps **88** umgewandelt hat. In beiden Fällen wird die Datentransaktion dem Client **74C** gegenüber als abgeschlossen anerkannt, bevor der Commit **90** die für die Festschreibung der Daten-

transaktion zur Speicherung im Festspeicher **78** notwendigen verzögerten Vorgänge abgeschlossen hat, und während die Datentransaktion immer noch im FSP 80-Speicherplatz resident ist. Als Folge hat ein Ausfall im FSP **80** oder des Dateiservers **72**, in dem der FSP **80** resident ist und der sich auf den FSP 80-Speicherplatz auswirkt, d.h. Speicher **76**, einen Verlust der Datentransaktion und aller mit der Datentransaktion in Zusammenhang stehenden Daten zum Ergebnis.

[0107] Weiterhin wurde ebenfalls in dieser Hinsicht in diesem Dokument weiter oben beschrieben, daß ein Dateiserver ein Transaktionsprotokoll zur Speicherung von Informationen über angeforderte Datentransaktionen, wie beispielsweise das Transaktionsprotokoll/Transaction Log (Log) **50L** von HAN File Server **10**, umfassen kann. Ein Transaktionsprotokoll speichert die Informationen in bezug auf jede Datentransaktion bzw. die zur Ausführung der Transaktion notwendige Zeit zur Ausführung der Transaktion, oder kann eine Aufzeichnung der gegenwärtigen und vergangenen Datentransaktionen speichern, und ermöglicht die erneute Ausführung gespeicherter Transaktionen. Dadurch bieten Transaktionsprotokolle Schutz vor dem Verlust von Datentransaktionen während der verzögerten Festschreibungsvorgänge für bestimmte Arten von Ausfällen, beispielsweise aufgrund eines Plattenlaufwerk 18-Ausfalls oder eines Fehlers bei den Festschreibungsvorgängen. Ein Ausfall im FSP **80** oder des Dateiservers **72**, in dem der FSP **80** resident ist, und der eine Auswirkung auf den FSP 80-Speicherplatz hat, kann jedoch auch in einem Verlust des Transaktionsprotokolls und damit der darin gespeicherten Datentransaktionen resultieren. Es sollte auch bemerkt werden, daß die Transaktionsprotokolle von Dateiserversystemen nach dem Stand der Technik typischerweise Darstellungen von Datentransaktionen auf einem relativ niedrigen Niveau der Dateiserverfunktionalität speichern, und zwar typischerweise unterhalb des FSOp 88-Betriebsniveaus und oftmals auf den Niveaus von Vorgängen, die durch Commit **90** ausgeführt wurden. Als solcher ist der Betrag an zu speichernden Informationen beträchtlich, und dementsprechend ist die Rekonstruktion und erneute Ausführung einer Datentransaktion schwierig und komplex. Außerdem, und weil Transaktionen auf einem niedrigen Niveau von Dateisystemvorgängen aufgezeichnet werden, erhöht sich die Latenzzeit des Dateiservers, d.h. die Verzögerungszeit, bevor eine Transaktion dem Client gegenüber anerkannt werden kann und zwecks fester Speicherung abgeschlossen wird, in dem Maße, in dem die Möglichkeit besteht, daß eine Datentransaktion aufgrund eines Ausfalls während des Aufzeichnungsprozesses verlorengeht.

[0108] Gemäß der vorliegenden Erfindung werden diese Probleme des Standes der Technik durch den Betrieb eines übergeordneten Transaktionsprotokollmechanismus, der in jedem der doppelten gleichrangigen Dateiserver resident ist und mit Transaktionsprotokoll-Spiegelungsmechanismen, die in dem entgegengesetzten gleichrangigen Dateiserver resident sind, über eine Hochgeschwindigkeits-Kommunikationsverbindung kommuniziert, vermieden. Dieser Transaktionsprotokoll- und Spiegelungsmechanismus ist in einem Dateiserversystem **70** durch Transaktionsprotokollmechanismen (TRLogs) **92L** und Transaktionsprotokoll-Spiegelungsmechanismen/Transaction Log Mirror Mechanisms (TLMirrors) **92M** ausgeführt, die in jedem der doppelten Dateiserver **72A** und **72B** realisiert sind, wobei jedes TRLog **92L** mit dem entsprechenden TRMirror **92M** in dem entgegengesetzten gleichrangigen Dateiserver **72** über die CPs **82** eines jeden der Dateiserver **72** und über CLink **84** kommuniziert. Die TLLogs **92L** und TLMirrors **92M** der vorliegenden Erfindung sind jeweils in einem HAN File Server **10** beispielhaft durch die zuvor beschriebenen Transaktionsprotokolle (Logs) **50L** dargestellt, die in JFiles **50** in den FEPs **44F** eines jeden der Blades **14** resident sind und mit den entsprechenden Protokoll-Spiegelungsmechanismen/Log Mirror Mechanisms (LMirrors) **54LA** und **54LB** kommunizieren, die in den BEPs **40B** der entgegengesetzten Blades **14** resident sind. Die CPs **82** und Clink **84** werden jeweils beispielhaft durch die BE Bus-Sys's **38O** dargestellt, die in den BEPs **44B** der Blades **14A** und **14B** resident sind und dort arbeiten, wobei Rechenblattelementschleifenbus **30** die Blades **14A** und **14B** untereinander verbindet. Weiterhin wird es in dieser Hinsicht von gewöhnlichen Fachleuten auf diesem Gebiet anerkannt werden, daß die vorliegende Erfindung in einem System mit einem einzelnen Dateiserver **72** realisiert werden kann, wobei der LMirror **54LA**, **54LB** in einer beliebigen anderen Domäne des Systems auf solch eine Art und Weise resident sein kann, daß er bei einem Ausfall des Dateiservers **72** funktionsfähig bleibt.

[0109] Wie in **Fig.** 4 veranschaulicht, weisen die TLLogs **92L** jeweils einen Protokollerzeuger (LGen) **92LG** zur Erzeugung von Protokolleinträgen (LEnts) **92LE** auf, die die angeforderten Datentransaktionen darstellen sowie einen Protokollspeicher/Log Store (LogS) **92LS** zum Speichern von LEnts **92LE**, wobei die Tiefe der LogSs **92S** von der Anzahl von aufzuzeichnenden Datentransaktionen abhängig ist, wie weiter unten abgehandelt werden wird. Die TLMirrors **92M** bestehen wiederum jeweils aus einem Spiegelspeicher/Mirror Store (MirrorS) **92MS** zum Speichern von LEnts **92LE**, die von dem entsprechenden TLLog **92L** und einem Spiegelmanager/Mirror Manager (MirrorM) **92MM** zum Speichern von LEnts **92LE** in dem MirrorS **92MS**, und zum Lesen von LEnts **92LE** von dem MirrorS **92MS** zur erneuten Ausführung durch den FPS **80**, von dem die LEnts **92LE** herstammen. Wie weiter unten beschrieben werden wird, kann bei alternativen Ausführungen der vorliegenden Erfindung der noch funktionsfähige der doppelten

Dateiserver **72** die Clients **74C** und CFiles **74F** des ausgefallenen Dateiservers **72** übernehmen. Bei diesen Ausführungen können die LEnts 92LE eines ausgefallenen Dateiservers **72** von dem TLMirror **92M** des noch funktionsfähigen Dateiservers **72** und die durch die LEnts **92LE** dargestellten Datentransaktionen gelesen, und durch den noch funktionsfähigen Dateiserver **72** erneut ausgeführt werden, um die Zustände der CFiles **74F** der übernommenen Clients **74C** des ausgefallenen Dateiservers **72** wiederherzustellen und zu rekonstruieren.

[0110] Wie in **Fig.** 4 dargestellt, überwacht der LGen 92LG eines jeden TLLog **92L** Informationen in bezug auf das obere Niveau von Dateiserver 72-Vorgängen und gewinnt sie, wie beispielsweise bei dem Anforderung 86-Betriebsniveau an einem Zwischenpunkt zwischen dem Anforderung 86-Betriebsniveau und dem Niveau, auf dem Anforderungen **86** in entsprechende Dateisystemvorgänge/File System Operations (FSOps) **88** umgewandelt werden, oder auf dem FSOps 88-Betriebsniveau. Wie in schematischer Form in **Fig.** 4 dargestellt, umfaßt daher jeder LEnt **92LE** ein Transaktionsfeld (TR) **94T**, das die Art der Datentransaktion identifiziert oder spezifiziert. Obwohl das System bei der gegenwärtig bevorzugten Ausführung auf dem Transaktionsniveau Clients nicht rückverfolgt, kann das System dies bei einer alternativen Ausführung und in solchen Fällen tun, und kann beispielsweise oder möglicherweise ein Client-Dateifeld (CF) **94C** umfassen, das CFile **74F**, zu dem die Datentransaktion gehört, und möglicherweise den die Datentransaktion anfordernden Client **74C**, identifizieren. Ein LEnt **92LE** umfaßt auch typischerweise ein Datenidentifikatorfeld (DI) **94I**, das die mit der Datentransaktion in Zusammenhang stehenden Daten bei einem Datenschreibvorgang identifiziert, d.h. eine Datenadresse oder andere Form eines Identifikators, und bei einer Datenschreibtransaktion ein Datenfeld (DA) **94D** umfassen kann, das eine Kopie der in CFile **74F** zu schreibenden Daten enthält. Es wird von gewöhnlichen Fachleuten auf diesem Fachgebiet verstanden werden, daß die Inhalte der LEnts **92LE** von Realisierung zu Realisierung variieren können, und beispielsweise von den in dem Dateiserversystem verwendeten spezifischen Dateisystemen, den unterstützten Vorgängen, dem spezifischen Niveau der Dateiserver 72-Vorgänge, auf dem LEnts 92LE erzeugt werden, usw. abhängig sein können. Die Inhalte von TR **94T** sind beispielsweise abhängig von dem Niveau des Dateiservers **72**, auf dem die mit den Datentransaktionen in Zusammenhang stehenden Informationen aus dem Betriebsablauf gewonnen werden, und können von den Inhalten des passenden Anforderung 86-Feldes abweichen, d.h. ein einzelner Befehls- oder Vorgangsbezeichner oder Anweisung sein, bis hin zu einer Folge oder Gruppe von Anweisungen oder Befehlen, die die zur Ausführung der Anforderung **86** notwendigen übergeordneten Vorgänge definieren oder identifizieren. Es ist jedoch notwendig und ausreichend, daß die Inhalte von LEnts **92LE** eine Rekonstruktion und Wiederherstellung jeder Datentransaktion ermöglichen. Es wird von gewöhnlichen Fachleuten auf diesem Fachgebiet jedoch geschätzt und gut verstanden werden, daß der Umfang und die Komplexität der einen LEnt **92LE** bildenden Informationen, die eine Datentransaktion definieren, bedeutend geringer als notwendig ist, um auf ähnliche Art und Weise eine Datentransaktion zu definieren, wenn sie von dem Betriebsfluß auf einem unteren Niveau gewonnen wurde, wie beispielsweise auf dem Commit 90- und Festspeicher 78-Niveau, wie dies bei Dateiserversystemen nach dem Stand der Technik typisch ist.

[0111] Wie in **Fig.** 4 angegeben, werden die durch den LGen **92LG** in dem residenten LogS **92LS** gespeicherten LEnts **92LE** auf herkömmliche Art und Weise gespeichert, und es wird verstanden werden, daß die Tiefe des LogS **92LS**, d.h. die Anzahl der LEnts **92LE**, die darin gespeichert sein können, von der Länge der zu erhaltenden Datentransaktionsaufzeichnungen abhängig ist. Im allgemeinen sollte die Tiefe von LEnts **92LE**, d.h. die Anzahl von LEnts **92LE**, die darin gespeichert sein können, ausreichend sein, um die maximale Anzahl von Datentransaktionen zu speichern, die während der maximalen Latenzzeit des Dateiservers **72** anfallen können, d.h. die maximale Zeitdauer, die zwischen dem Empfang einer Anforderung **86** und dem Abschluß der Festschreibung der Datentransaktion zur Speicherung im Festspeicher **78** auftreten kann.

[0112] Zusätzlich, und gemäß der vorliegenden Erfindung, wird jedoch bei jedem der Dateiserver **72A** und **72B** jeder als Reaktion auf eine empfangene Anforderung **86** durch den LGen **92LG** erzeugter LEnt **92LE** über die aus den CPs **82** eines jeden Dateiservers **72** und CLink **84** bestehende Hochgeschwindigkeits-Kommunikationsverbindung zu dem TLMirror **92M** in dem anderen Dateiserver **72** übertragen, wobei der MirrorM **92MM** des den LEnt **92LE** empfangenden TLMirror **92M** den LEnt **92LE** in dem MirrorS **92MS** speichert. Erneut sollte die Tiefe von MirrorS **92MS** ausreichend sein, um die maximale Anzahl von Datentransaktionen zu speichern, d.h. die maximale Zeitdauer, die zwischen dem Empfang einer Anforderung **86** und dem Abschluß der Festschreibung der Datentransaktion zur Speicherung im Festspeicher **78** auftreten kann. Es wird jedoch anerkannt werden, daß die Tiefe des MirrorS **92MS** je nach Abhängigkeit von der Länge der gewünschten Datentransaktionsaufzeichnung größer oder kleiner sein kann.

[0113] Bei der gegenwärtig bevorzugten Ausführung der Erfindung wird die positive Rückmeldung des Empfangs und die Annahme einer Datentransaktion an den Client **74C**, von dem die Anforderung **86** herstammt, nicht direkt durch den FSP **80** erzeugt,

der die Anforderung **86** empfängt. Stattdessen sendet der MirrorM **92MM** eine positive Rückmeldung an den LGen **92LG** zurück, der die Quelle des LEnt **92LE** war, nachdem der TLMirror **92M** einen LEnt **92LE** empfangen und gespeichert hat, wobei der LGen **92LG** den FSP **80** über die positive Rückmeldung benachrichtigt. Der FSP **80** erzeugt dann eine entsprechende positive Rückmeldung an den entsprechenden Client **74C**, daß die Datentransaktion angenommen und abgeschlossen wurde.

[0114] Es wird erkannt werden, daß die Verzögerung bei der Erzeugung einer positiven Rückmeldung an den Client **74C** etwas größer sein kann, als die für den FSP **80** zur Erzeugung einer positiven Rückmeldung direkt durch, beispielsweise die Übertragungszeit des LEnt **92LE** an den MirrorM **92MM**, und die Übertragungszeit der zurückgesendeten positiven Rückmeldung. Es wird jedoch erkannt werden, daß die zur sicheren Speicherung in dem TLMirror **92M** des einer Anforderung **86** entsprechenden LEnt **92LE** erforderliche Latenzzeit über die Hochgeschwindigkeitskommunikationsverbindung zwischen den Dateiservern **72** typischerweise bedeutend geringer ist als die für die Festschreibung der Datentransaktion zur Speicherung im Festspeicher **78** erforderliche Latenzzeit. Die zur sicheren Speicherung eines auf einem niedrigeren Niveau in einem FSP **80** in einem LogS **92LS** erzeugten LEnt **92LE** erforderliche Latenzzeit wie bei den Systemen nach dem Stand der Technik, reduziert daher die Anfälligkeitszeitdauer, in der eine Datentransaktion aufgrund eines Ausfalls in dem FSP **80** verlorengehen kann.

[0115] Es wird jedoch auch erkannt werden, daß die zur sicheren Speicherung in dem LMirror **92M** des einer Anforderung **86** entsprechenden LEnt **92LE** erforderliche Latenzzeit über die Hochgeschwindigkeitskommunikationsverbindung zwischen den Dateiservern **72** durch die Latenzzeit der normalen Eingabe-Ausgabeverarbeitung des Dateisystems maskiert wird, solange die Übertragungsgeschwindigkeit der Hochgeschwindigkeitskommunikationsverbindung ausreichend hoch ist. Bei dem gegenwärtigen beispielhaften HAN File Server **10** besteht diese Verbindung durch eine Optikfaserverbindung, wobei aber die tatsächliche Geschwindigkeit der Kommunikationsverbindung nicht kritisch ist, solange diese Anforderung erfüllt wird. Als solche ist die Latenzzeit der TLMirror 92M-Vorgänge für einen Client **74C** nicht sichtbar, weil sie durch die Datentransaktionsverarbeitungszeit des FSP **80** maskiert ist, wobei die Dateiserver **72** keine offensichtliche Leistungseinbuße durch den Transaktionsprotokollmechanismus der vorliegenden Erfindung erleiden.

[0116] Schließlich wird verstanden werden, daß, wie weiter oben beschrieben, der TLMirror **92M** bei einem vorgegebenen Dateiserver **72** mindestens die nicht abgeschlossenen, d.h. nicht festgeschriebenen, in dem Dateiserver **72** nach Eintritt eines Ausfalls in dem Dateiserver **72** anhängigen Datentransaktionen erhält. Danach, und nachdem die Mechanismen des ausgefallenen Dateiservers **72** den Status und den Betrieb des ausgefallenen Dateiservers **72** wiederherstellen, wie in diesem Dokument oben in bezug auf den HAN File Server **10** beschrieben, liest der TLMirror **92M** die gespeicherten LEnts **92LE** aus dem MirrorS **92MS**, und zur erneuten Ausführung in den wiederhergestellten Dateiserver **72** zurück. Die erneute Ausführung der durch die LEnts **92LE** dargestellten und definierten Datentransaktionen stellt deshalb den Status der von dem ausgefallenen Dateiserver **72** unterstützten CFiles **74F** wieder in den Zustand her, der von den Clients **74C** erwartet wird, wobei von diesem Punkt an der Normalbetrieb wieder aufgenommen wird.

[0117] Bei alternativen Ausführungen, und wie weiter oben in bezug auf den beispielhaft dargestellten HAN File Server **10** beschrieben, kann der noch funktionsfähige von doppelten Dateiservern **72** die Clients **74C** und CFiles **74F** des ausgefallenen Dateiservers **72** durch die in bezug auf den HAN File Server **10** beschriebenen Failover-Mechanismen übernehmen. Bei diesen Ausführungen werden die durch den ausgefallenen Dateiserver **72** unterstützten Kommunikationsverbindungen zu den Clients **74C** auf den noch funktionsfähigen Dateiserver **72** übertragen, wie auch die CFiles **74F** des ausgefallenen Dateiservers **72**. Der FSP **80** des noch funktionsfähigen Dateiservers **72** liest dann die LEnts **92LE** aus dem ausgefallenen Dateiserver **72** von dem TLMirror **92M**, und die durch die LEnts **92LE** dargestellten Datentransaktionen werden durch den noch funktionsfähigen Dateiserver **72** erneut ausgeführt, um den Status der CFiles **74F** der übernommenen Clients **74C** des ausgefallenen Dateiservers **72** wiederherzustellen.

[0118] Es wird gewöhnlichen Fachleuten auf diesem Fachgebiet offensichtlich sein, daß die vorliegende Erfindung bei jeder Form von gemeinsam genutzten Ressourcen realisiert werden kann, bei der zuverlässige Kommunikationen zwischen Clients sowie die Erhaltung und die Wiederherstellung von Daten oder betrieblichen Transaktionen erforderlich ist, wie beispielsweise bei Kommunikationsservern, verschiedenen Arten von Datenprozessorservern, Druckservern usw., sowie bei dem in diesem Dokument als Beispiel verwendeten Dateiserver. Es wird auch offensichtlich sein, daß die vorliegende Erfindung auf ähnliche Art und Weise bei anderen Realisierungen angepaßt und realisiert werden kann, bei denen Dateiserver zur Anwendung kommen, die beispielsweise unterschiedliche RAID-Technologien, unterschiedliche Speichertechnologien, unterschiedliche Kommunikationstechnologien und andere Informationsverarbeitungsverfahren und -techniken, wie

beispielsweise die Bildverarbeitung, verwenden. Die Anpassung der vorliegenden Erfindung an unterschiedliche Formen gemeinsam genutzter Ressourcen, an unterschiedliche Ressourcenverwalter, unterschiedliche Systemkonfigurationen und -strukturen sowie unterschiedliche Protokolle wird für gewöhnliche Fachleute auf diesem Fachgebiet offensichtlich sein.

Patentansprüche

1. Übergeordneter Transaktionsprotokollierungsmechanismus zur Verwendung in Verbindung mit einer gemeinsam genutzten Systemressource (**10**) mit einem Ressourcen-Subsystem (**12**) zur Ausführung von untergeordneten Systemressourcenvorgängen und einem Steuerungs-/Verarbeitungssubsystem (**14**) mit einer ersten und einer zweiten Sub-Rechnereinheit (**14A**,**14B**), die jeweils einen Systemressourcenprozessor (**80A**,**80B**) aufweisen, der übergeordnete Systemressourcenvorgänge ausführt und Systemressourcenanforderungen von Clients (**34C**,**74C**) in entsprechende untergeordnete Systemressourcenvorgänge umwandelt, der in der ersten und in der zweiten Sub-Rechnereinheit (**14A**,**14B**) des Steuerungs-/Verarbeitungssubsystems (**14**) jeweils folgendes aufweist:
– einen in der Sub-Rechnereinheit (**14A**,**14B**) angeordneten Protokollerzeuger (**50G**,**92LG**) zur Gewinnung von übergeordneten Vorgangsinformationen in Bezug auf jeden übergeordneten Vorgang der Sub-Rechnereinheit (**14A**,**14B**); und
– ein in der Sub-Rechnereinheit (**14A**,**14B**) angeordnetes Transaktionsprotokoll zur Speicherung der übergeordneten Vorgangsinformationen, wobei der protokollierende Mechanismus zur Wiederherstellung des Betriebes der Sub-Rechnereinheit (**14A**,**14B**) nach einem Ausfall der Sub-Rechnereinheit (**14A**,**14B**) für das Lesen der übergeordneten Vorgangsinformationen aus dem Transaktionsprotokoll und das Wiederherstellen des Ausführungsstatus der Sub-Rechnereinheit (**14A**,**14B**) verantwortlich ist; und
– ein Transaktionsprotokoll-Spiegelungsmechanismus, der in der anderen Sub-Rechnereinheit (**14A**,**14B**) angeordnet ist und mit dem Protokollerzeuger (**50G**,**92LG**) der Sub-Rechnereinheit (**14A**,**14B**) zum Empfang und zur Speicherung der Spiegelungskopien der übergeordneten Vorgangsinformation der Sub-Rechnereinheit (**14A**,**14B**) kommuniziert, wobei der Transaktionsprotokoll-Spiegelungsmechanismus zur Wiederherstellung des Betriebes der Sub-Rechnereinheit (**14A**,**14B**) nach deren Ausfall für das Lesen der Spiegelungskopien des übergeordneten Vorgangs der Sub-Rechnereinheit (**14A**,**14B**) aus dem in der anderen Sub-Rechnereinheit (**14A**,**14B**) angeordneten Transaktionsprotokoll-Spiegelungsmechanismus in die Sub-Rechnereinheit (**14A**,**14B**) und für die Wiederherstellung des Ausführungsstatus der Sub-Rechnereinheit (**14A**,**14B**) verantwortlich ist.

2. Transaktionsprotokollierungsmechanismus nach Anspruch 1, dadurch gekennzeichnet, dass die übergeordnete Vorgangsinformation in Bezug auf jede an die erste und an die zweite Sub-Rechnereinheit (**14A**,**14B**) gerichtete Systemressourcenanforderung vor Abschluss der entsprechenden Ressourcenanfrage durch die Sub-Rechnereinheit (**14A**,**14B**) gewonnen wird, und wobei eine Client-Systemressourcenanforderung durch das Ressourcen-Subsystem (**12**) als angenommen bestätigt wird, nachdem die übergeordnete Vorgangsinformation in dem Transaktionsprotokoll und in dem Transaktionsprotokoll-Spiegelungsmechanismus der Sub-Rechnereinheit (**14A**,**14B**) gespeichert sind.

3. Transaktionsprotokollierungsmechanismus nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass die gemeinsam genutzte Systemressource als Dateiserver (**10**) ausgebildet ist, der Clients (**34C**,**74C**) Dateisystemanteile zur gemeinsamen Nutzung bereitstellt, wobei die gemeinsam genutzten Ressourcenvorgänge Dateilese- und -schreibvorgänge aufweisen.

4. Verfahren zur Protokollierung von Systemressourcentransaktionen und zur Wiederherstellung des Ausführungsstatus von Systemressourcenanforderungen in einer Clients (**34C**,**74C**) Systemressourcendienste bietenden Systemressource (**10**) mit gemeinsamer Nutzung als Reaktion auf Systemressourcenanforderungen durch die Clients (**34C**,**74C**), wobei die Systemressource (**10**) ein Ressourcensubsystem (**12**) zur Ausführung untergeordneter Systemressourcenvorgänge und ein Steuerungs-/Verarbeitungs-Subsystem (**14**) mit ersten und zweiten Sub-Rechnereinheiten (**14A**,**14B**) aufweist, wobei jede Sub-Rechnereinheit (**14A**,**14B**) einen Systemressourcenprozessor (**80A**,**80B**), der übergeordnete Systemressourcenvorgänge ausführt und Systemressourcenanforderungen von Clients (**34C**,**74C**) in entsprechende untergeordnete Systemressourcenvorgänge umwandelt und einen Transaktionsprotokollierungsmechanismus umfasst, das in jeder Sub-Rechnereinheit (**14A**,**14B**) des Steuerungs-/Verarbeitungssubsystems (**14**) die folgenden Schritte durchführt:
– Gewinnen von übergeordneten Vorgangsinformationen in Bezug auf jeden übergeordneten Vorgang der Sub-Rechnereinheit (**14A**,**14B**); und
– Speichern der übergeordneten Vorgangsinformationen in einem in der Sub-Rechnereinheit (**14A**,**14B**) angeordneten Transaktionsprotokoll; und
– Lesen der übergeordneten Vorgangsinformationen aus dem in der Sub-Rechnereinheit (**14A**,**14B**) angeordneten Transaktionsprotokoll nach einer Wiederherstellung des Betriebes der Sub-Rechnereinheit nach deren Ausfall und Wiederherstellen des Ausführungsstatus der Sub-Rechnereinheit (**14A**,**14B**); und

– Speichern von Spiegelungskopien der übergeordneten Vorgangsinformationen, die von dem Protokollerzeuger (**50G**,**92LG**) erzeugt wurden in einen in der anderen Sub-Rechnereinheit (**14A**,**14B**) angeordneten Transaktionsprotokoll-Spiegelungsmechanismus; und
– als Reaktion auf die Wiederherstellung des Betriebes der Sub-Rechnereinheit (**14A**,**14B**) nach deren Ausfall das Lesen der Spiegelungskopien des übergeordneten Vorgangs, der in dem in der anderen Sub-Rechnereinheit (**14A**,**14B**) angeordneten Transaktionsprotokoll-Spiegelungsmechanismus angeordnet ist, in die Sub-Rechnereinheit (**14A**,**14B**) und Wiederherstellen des Ausführungsstatus der anderen Sub-Rechnereinheit (**14A**,**14B**).

5. Verfahren zur Protokollierung von Systemressourcentransaktionen und Wiederherstellung des Ausführungsstatus von Systemressourcenanforderungen nach Anspruch 4, dadurch gekennzeichnet, dass die übergeordneten Vorgangsinformationen in Bezug auf jede an die Sub-Rechnereinheit (**14A**,**14B**) gerichtete Systemressourcenanforderung vor Abschluss der entsprechenden Ressourcenanforderung durch die Sub-Rechnereinheit (**14A**,**14B**) gewonnen wird, wobei eine Client-Systemressourcenanforderung durch das Ressourcensubsystem (**12**) als angenommen bestätigt wird, nachdem die übergeordneten Vorgangsinformationen im Transaktionsprotokoll und in dem Transaktionsprotokoll-Spiegelungsmechanismus, der Sub-Rechnereinheit (**14A**,**14B**) gespeichert sind.

6. Verfahren zur Protokollierung von Systemressourcentransaktionen und Wiederherstellung des Ausführungsstatus von Systemressourcenanforderungen nach Anspruch 4 oder 5, dadurch gekennzeichnet, dass die gemeinsam genutzte Systemressource als Dateiserver (**10**) ausgebildet ist, der Clients (**34C**,**74C**) Dateisystemanteile zur gemeinsamen Nutzung bereitstellt, wobei die gemeinsam genutzten Ressourcenvorgänge Dateilese- und -schreibvorgänge aufweisen.

Es folgen 4 Blatt Zeichnungen

Anhängende Zeichnungen

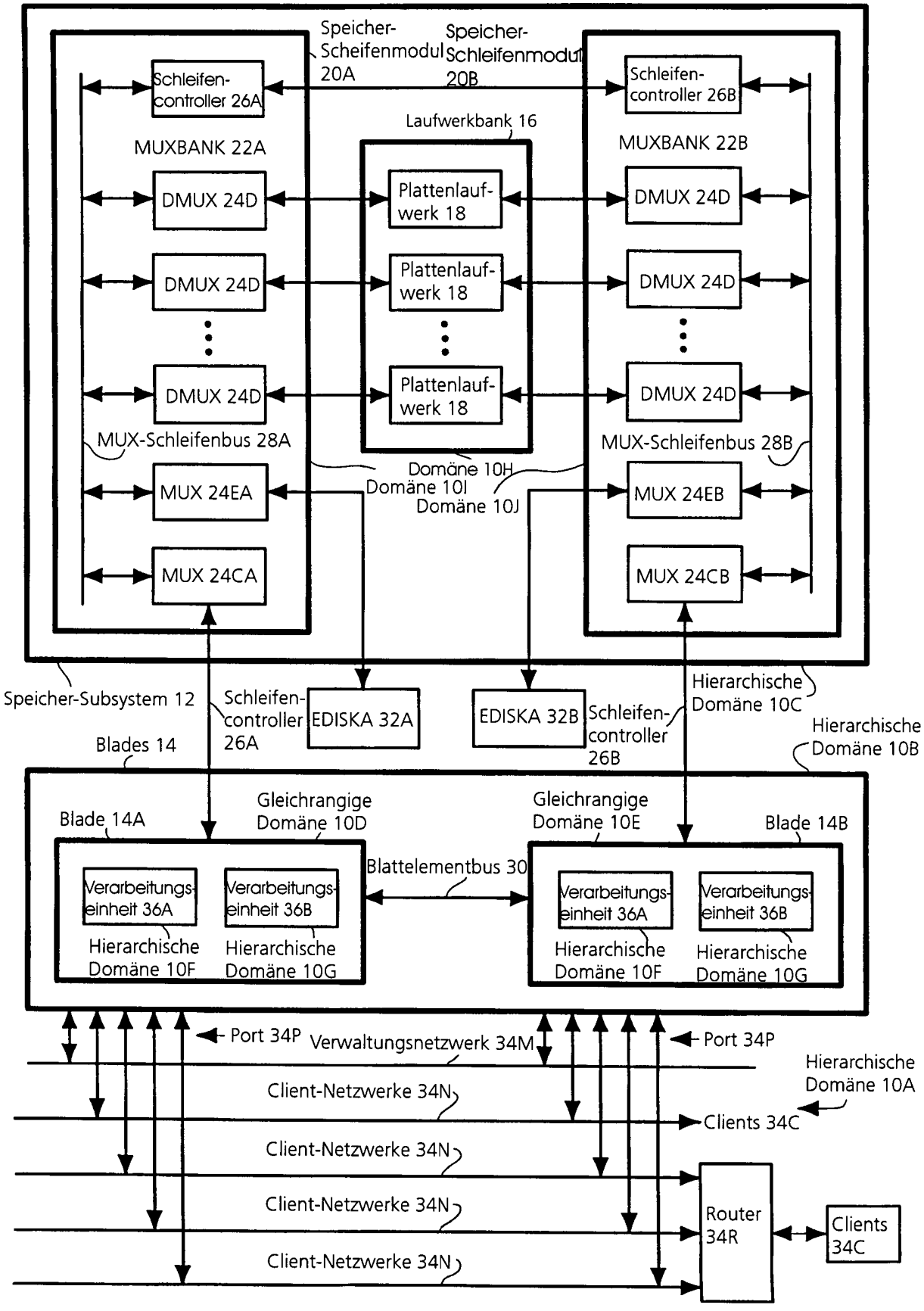


Fig. 1

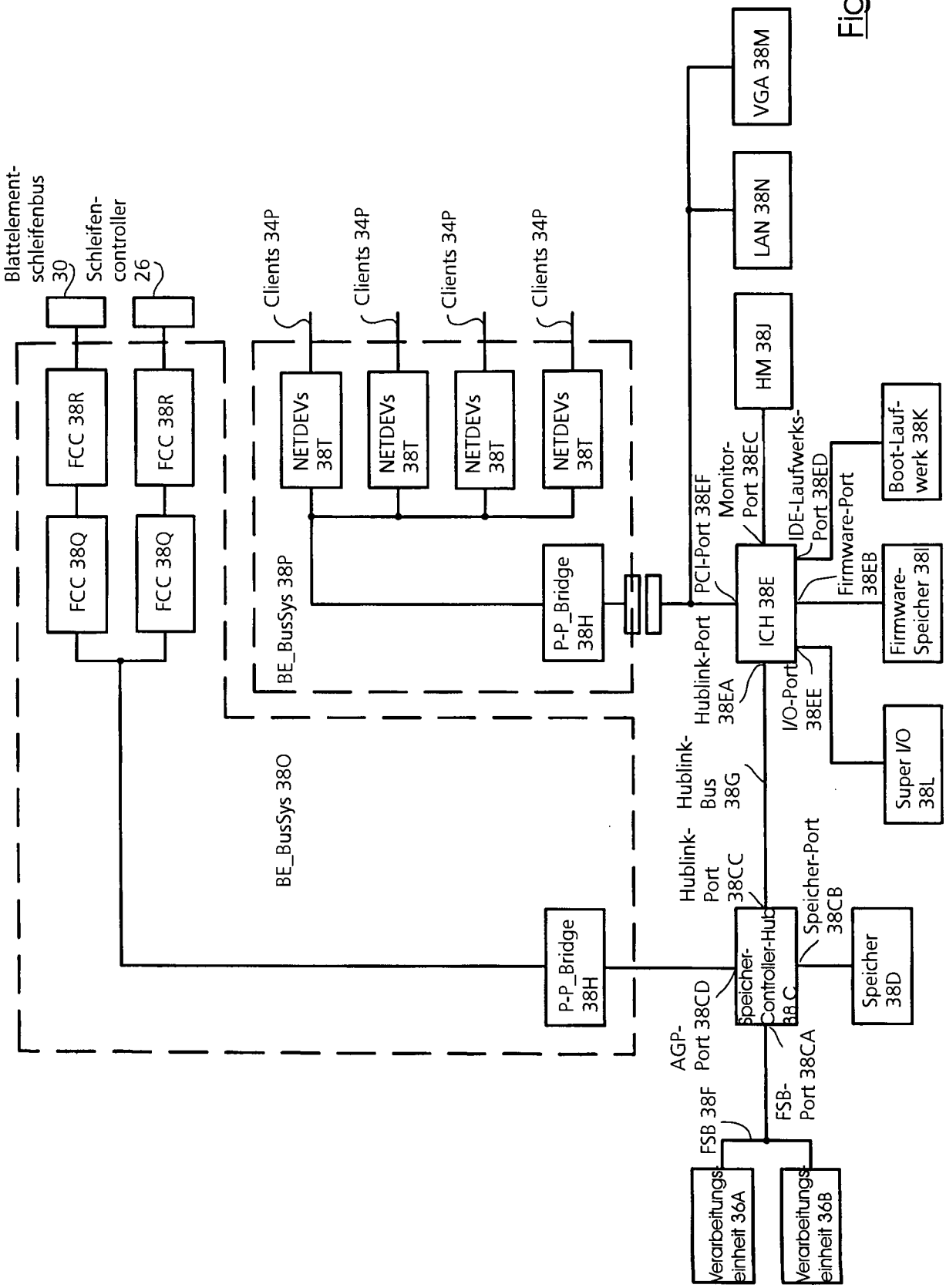

Blattelement-schleifenbus 30
Schleifen-controller 26
FCC 38R
FCC 38Q
FCC 38R
FCC 38Q
Clients 34P
Clients 34P
Clients 34P
Clients 34P
NETDEVs 38T
NETDEVs 38T
NETDEVs 38T
NETDEVs 38T
BE_BusSys 38P
BE_BusSys 38O
P-P_Bridge 38H
P-P_Bridge 38H
VGA 38M
LAN 38N
HM 38J
PCI-Port 38EF
Monitor-Port 38EC
IDE-Laufwerks-Port 38ED
Firmware-Port 38EB
Boot-Laufwerk 38K
ICH 38E
Firmware-Speicher 38I
Hublink-Port 38EA
I/O-Port 38EE
Super I/O 38L
Hublink-Bus 38G
Hublink-Port 38CC
Speicher-Port 38CB
Speicher-Controller-Hub 38C
Speicher 38D
AGP-Port 38CD
FSB 38F
FSB-Port 38CA
Verarbeitungseinheit 36A
Verarbeitungseinheit 36B

Fig. 2

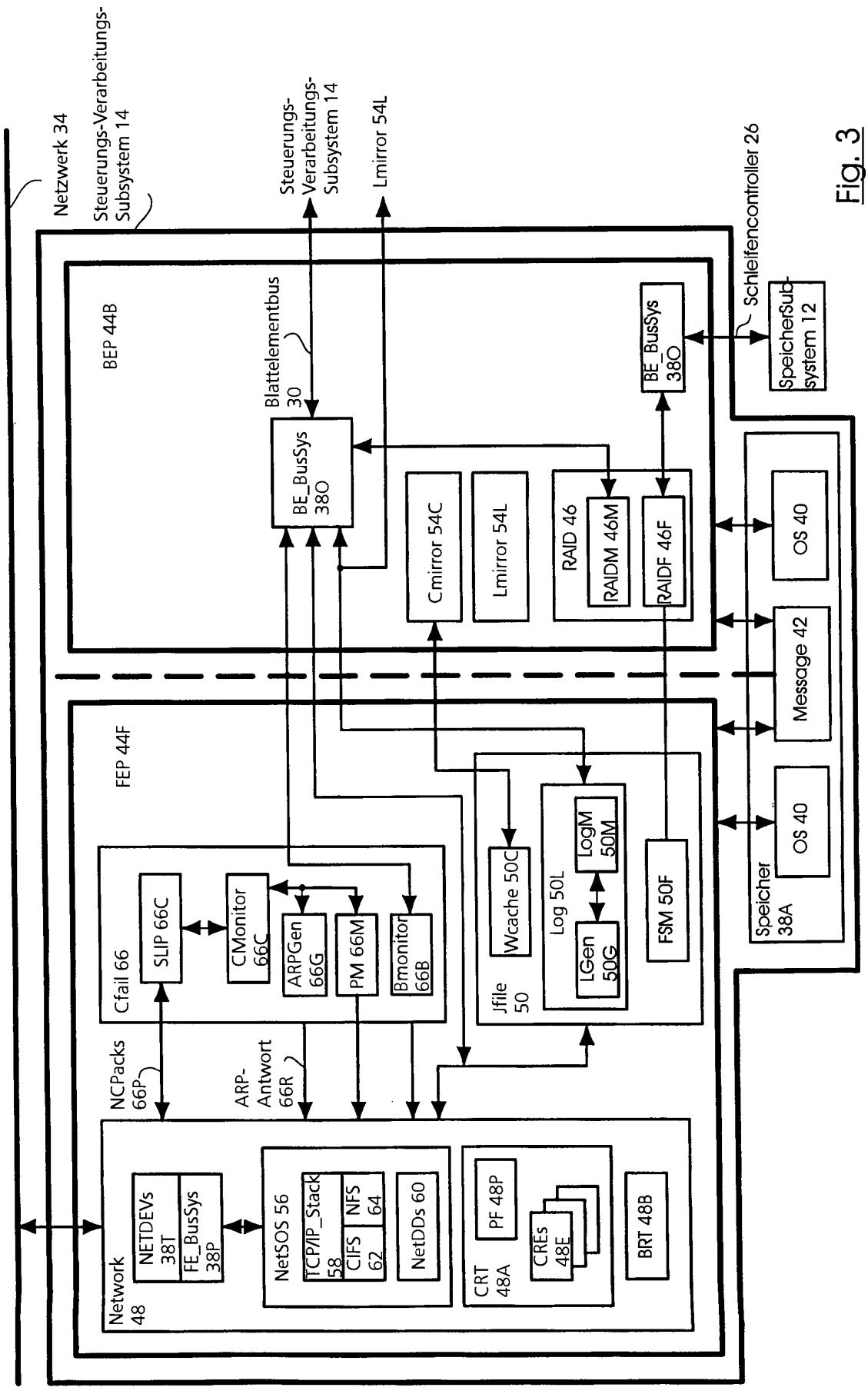

Netzwerk 34
Steuerungs-Verarbeitungs-Subsystem 14
Steuerungs-Verarbeitungs-Subsystem 14
Lmirror 54L
Schleifencontroller 26
BEP 44B
Blattelementbus 30
BE_BusSys 38O
BE_BusSys 38O
SpeicherSubsystem 12
Cmirror 54C
Lmirror 54L
RAID 46
RAIDM 46M
RAIDF 46F
OS 40
Message 42
FEP 44F
Cfail 66
SLIP 66C
CMonitor 66C
ARPGen 66G
PM 66M
Bmonitor 66B
Jfile 50
Wcache 50C
Log 50L
LogM 50M
LGen 50G
FSM 50F
Speicher 38A
OS 40
NCPacks 66P
ARP-Antwort 66R
Network 48
NETDEVs 38T
FE_BusSys 38P
NetSOS 56
TCP/IP_Stack 58
CIFS 62
NFS 64
NetDDs 60
CRT 48A
PF 48P
CREs 48E
BRT 48B

Fig. 3

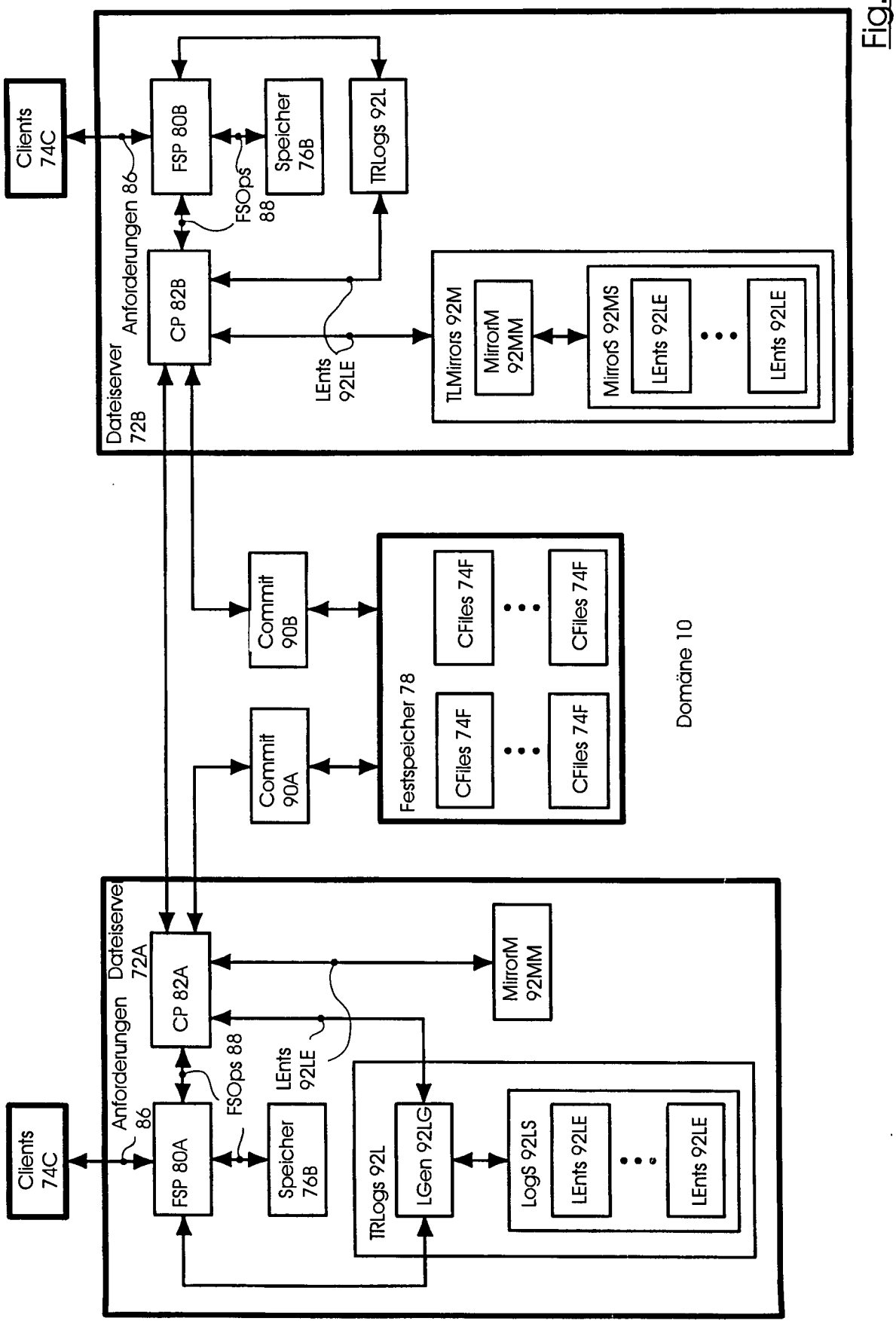

Clients 74C
FSP 80B
Speicher 76B
TRLogs 92L
Anforderungen 86
FSOps 88
CP 82B
Dateiserver 72B
LEnts 92LE
TLMirrors 92M
MirrorM 92MM
MirrorS 92MS
LEnts 92LE
LEnts 92LE
Commit 90B
Commit 90A
Festspeicher 78
CFiles 74F
CFiles 74F
CFiles 74F
CFiles 74F
Domäne 10
Clients 74C
FSP 80A
Speicher 76B
Anforderungen 86
FSOps 88
CP 82A
Dateiserver 72A
LEnts 92LE
MirrorM 92MM
TRLogs 92L
LGen 92LG
LogS 92LS
LEnts 92LE
LEnts 92LE

Fig. 4