(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4609996号
(P4609996)

(45)発行日　平成23年1月12日(2011.1.12)　(24)登録日　平成22年10月22日(2010.10.22)

(51)Int.Cl.		F I		
G06F 13/10	(2006.01)	G06F	13/10	340A
G06F 12/00	(2006.01)	G06F	12/00	545A
G06F 3/06	(2006.01)	G06F	12/00	514E
		G06F	3/06	301A

請求項の数 15　(全 22 頁)

(21)出願番号	特願2004-561479 (P2004-561479)
(86)(22)出願日	平成15年11月25日 (2003.11.25)
(65)公表番号	特表2006-510976 (P2006-510976A)
(43)公表日	平成18年3月30日 (2006.3.30)
(86)国際出願番号	PCT/EP2003/050895
(87)国際公開番号	WO2004/057798
(87)国際公開日	平成16年7月8日 (2004.7.8)
審査請求日	平成18年10月30日 (2006.10.30)
(31)優先権主張番号	02102852.7
(32)優先日	平成14年12月20日 (2002.12.20)
(33)優先権主張国	欧州特許庁 (EP)

(73)特許権者　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国１０５０４　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
(74)代理人　100108501
弁理士　上野　剛史
(74)代理人　100112690
弁理士　太佐　種一
(74)代理人　100091568
弁理士　市位　嘉宏

最終頁に続く

(54)【発明の名称】非トラステッド・サーバ環境におけるＳＡＮ管理のためのセキュア・システムおよび方法

(57)【特許請求の範囲】

【請求項１】

ストレージ・エリア・ネットワーク（ＳＡＮ）を操作するためのシステムであって、

ＳＡＮ管理サーバと、

トラステッド・オペレーティング・システム（ＯＳ）を実行するＳＡＮ管理クライアントと、

非トラステッドＯＳとリモート・アクセス・サーバ機能を実行する少なくとも１つのコンピュータ・システムと

を備えており、

前記ＳＡＮ管理クライアントは、ファイバ・チャネル（ＦＣ）・アダプタを介してＳＡＮ及び前記少なくとも１つのコンピュータ・システムと接続されており、

前記少なくとも１つのコンピュータ・システムは、前記ＦＣアダプタを介して及び前記ＳＡＮへのアクセスを制御するためのファイアウォールを介してＳＡＮに結合されており、

前記ＳＡＮ管理クライアントが、前記非トラステッドＯＳのそれぞれの代わりに、前記ファイバ・チャネル・ネットワークにコマンドを発行するように構成されており、

前記トラステッドＯＳ上で実行されるファイアウォール制御アプリケーションのみが特定のコマンドを前記非トラステッドＯＳに転送するかどうかをファイアウォールに指示するために当該ファイアウォール設定を変更することが可能である、前記システム。

【請求項２】

前記ＳＡＮ管理サーバが、第１のコマンド・セットと第２のコマンド・セットとを分離するように構成されており、前記第１のコマンド・セットが前記ＳＡＮとともに前記ＳＡＮ管理クライアントによって処理され、前記第２のコマンド・セットが前記ＳＡＮへのアクセスなしに前記非トラステッドＯＳによって処理される、請求項１に記載のシステム。

【請求項３】

前記ＳＡＮ管理クライアントが、第１のコマンド・セットと第２のコマンド・セットとを分離するように構成されており、前記第１のコマンド・セットが前記ＳＡＮとともに前記ＳＡＮ管理クライアントによって処理され、前記第２のコマンド・セットが前記ＳＡＮへのアクセスなしに前記非トラステッドＯＳによって処理される、請求項１又は２に記載のシステム。

【請求項４】

前記ＦＣアダプタが、前記ＳＡＮ管理クライアントを認証するように構成されている、請求項１又は２に記載のシステム。

【請求項５】

冗長性を提供するために、バックアップ用の１つまたは複数のＳＡＮ管理クライアントを備えている、請求項１～３のいずれか一項に記載のシステム。

【請求項６】

前記ＳＡＮ管理クライアントのみが前記ＳＡＮからのメッセージを受信するために登録されており、前記ＳＡＮ管理クライアントが、前記ＳＡＮ管理サーバにのみ前記メッセージを転送するように構成されている、請求項１～４のいずれか一項に記載のシステム。

【請求項７】

前記ＳＡＮ管理サーバが、前記リモート・アクセス・サーバ機能にアクセスするための許可データを保持するためのリポジトリを備えている、請求項１に記載のシステム。

【請求項８】

前記ＳＡＮ管理クライアントが、前記リモート・アクセス・サーバ機能にアクセスするための許可データを保持するためのリポジトリを備えている、請求項１に記載のシステム。

【請求項９】

前記ＳＡＮ管理クライアントおよび前記ＦＣアダプタが、各非トラステッドＯＳによるリソースの使用を要求するために使用される情報を収集するように構成されている、請求項１～８のいずれか一項に記載のシステム。

【請求項１０】

前記ＳＡＮ管理サーバが、アクセス権を設定するためにファイアウォール制御アプリケーションと通信するように適合されている、請求項１～９のいずれか一項に記載のシステム。

【請求項１１】

前記リモート・アクセス・サーバ機能が、Ｔｅｌｎｅｔ／ｓｓｈｄサーバ機能で実現される、請求項１に記載のシステム。

【請求項１２】

ＳＡＮ管理サーバと、トラステッド・オペレーティング・システム（ＯＳ）を実行するＳＡＮ管理クライアントと、非トラステッドＯＳとリモート・アクセス・サーバ機能を実行する少なくとも１つのコンピュータ・システムとを備えているストレージ・エリア・ネットワーク（ＳＡＮ）において、当該ＳＡＮを操作させるための方法であって、前記ＳＡＮ管理クライアントは、ファイバ・チャネル（ＦＣ）・アダプタを介してＳＡＮ及び前記少なくとも１つのコンピュータ・システムと接続されており、前記少なくとも１つのコンピュータ・システムは、前記ＦＣアダプタを介して及び前記ＳＡＮへのアクセスを制御するためのファイアウォールを介してＳＡＮに結合されており、

前記方法は、

前記ＳＡＮ管理サーバによって発行された要求を少なくとも２つのグループに分離するステップと、

非トラステッドＯＳの代わりに前記ＦＣアダプタおよび前記ＳＡＮによって第１のグループの要求が処理されるステップと、
前記ＦＣアダプタおよび前記ＳＡＮとの間で要求を送受信する必要なしに前記非トラステッドＯＳによって第２のグループの要求が処理されるステップと、
前記トラステッドＯＳ上で実行されるファイアウォール制御アプリケーションのみが要求を前記非トラステッドＯＳに転送するかどうかをファイアウォール設定を変更することによって可能にするために、当該ファイアウォールに指示するステップと
を含む、前記方法。
【請求項１３】
前記第１のグループからの要求を実行するために前記ＳＡＮ管理クライアントに許可データを前記ＳＡＮ管理サーバによって提供するステップをさらに実行させる、請求項１２に記載の方法。
【請求項１４】
前記第２のグループからの要求を実行するために前記非トラステッドＯＳに許可データを前記ＳＡＮ管理サーバ及びＳＡＮ管理クライアントによって提供するステップをさらに実行させる、請求項１２又は１３に記載の方法。
【請求項１５】
コンピュータに請求項１２～１４のいずれか一項に記載の方法の各ステップを実行させるためのコンピュータ・プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、一般に、ストレージ・エリア・ネットワーク（storagearea network）に関する。特に、本発明は、複数サーバが１つのファイバ・チャネル・アダプタを共用するサーバ環境においてストレージ・エリア・ネットワークを操作するための方法およびシステムに関する。
【背景技術】
【０００２】
ファイバ・チャネルは、数十キロメートル離すことができる入出力（Ｉ／Ｏ）装置およびホスト・システムを相互接続するために使用する高速の全二重（full-duplex）シリアル通信技術である。これは、ＳＣＳＩおよびＰＣＩに見られるスループットおよび信頼性のような従来の入出力インターフェースの最良の特徴と、イーサネットおよびトークリングに見られる接続性およびスケーラビリティのようなネットワーキング・インターフェースの最良の特徴を組み合わせるものである。これは、既存のコマンドの送達のためのトランスポート・メカニズムを提供し、かなりの量の処理をハードウェアで実行できるようにすることによりハイパフォーマンスを達成するアーキテクチャを提供する。これは、ＳＣＳＩおよびＩＰのようなレガシー・プロトコルおよびドライバでも機能することができ、既存のインフラストラクチャに容易に導入できるようにするものである。
【０００３】
ファイバ・チャネルは、情報のソースとユーザとの間でその情報を転送する。この情報としては、コマンド、制御、ファイル、グラフィックス、ビデオ、およびサウンドを含むことができる。ファイバ・チャネル接続は、入出力装置に存在するファイバ・チャネル・ポートと、ホスト・システムと、それらを相互接続するネットワークとの間に確立される。ネットワークは、ファイバ・チャネル・ポートを相互接続するために使用される交換機、ハブ、ブリッジ、および中継器のような諸要素から構成される。
【０００４】
ファイバ・チャネル・アーキテクチャでは、３通りのファイバ・チャネル・トポロジが定義されている。これらは、２地点間（Point-to-Point）、スイッチ・ファブリック（Switched Fabric）、およびアービトレーテッド・ループ（ArbitratedLoop）である。
【０００５】

ファイバ・チャネル・スイッチ（またはスイッチ・ファブリック）は、一般にゾーニングと呼ばれる機能も含む。この機能により、ユーザはスイッチ・ポートを複数のポート・グループに区分することができる。あるポート・グループまたはゾーン内のポートは、同じポート・グループ（ゾーン）内の他のポートのみと通信することができる。ゾーニングを使用することにより、あるグループのホストおよび装置からの入出力は任意の他のグループの入出力から完全に分離することができ、したがって、グループ間の干渉の可能性を防止する。

【０００６】

これは、「ソフト・ゾーニング（soft zoning）」とも呼ばれる。このソフト・ゾーニングが機能する方法は、ユーザがノードのワールド・ワイド名（World WideName）、すなわち、ワールド・ワイド・ポート名（World Wide Port Name：ＷＷＰＮ）またはワールド・ワイド・ノード名（World WideNode Name：ＷＷＮＮ）に応じて、１つのゾーンに複数ノードを割り当てることである。ネーム・サーバはこの情報を収集するものであり、これは交換機内に埋め込まれた機能である。その場合、あるポートがどのノードに接続可能であるかを見つけるためにネーム・サーバと通信すると、ネーム・サーバは必ず、そのポートのゾーン内にあるノードのみで応答することになる。

【０００７】

標準のファイバ・チャネル・デバイス・ドライバはこのようにネーム・サーバと通信するので、このタイプのゾーニングはほとんどの状況に適切である。しかし、許可された接続のリストに含まれないノードへのアクセスを試みると思われるデバイス・ドライバが設計されることはあり得ることである。これが発生した場合、交換機はその違反を防止することも検出することもないであろう。

【０００８】

このケースを防止するために、交換機は任意選択で、ソフト・ゾーニングに加えて「ハード・ゾーニング（hard zoning）」と呼ばれるメカニズムも実現し、その場合、交換網は、各フレームのソース・アドレスと宛先アドレスのみに基づいて、このフレームのトランスポートが許可されるかどうかを決定する。

【０００９】

ファイバ・チャネル・ストレージ・エリア・ネットワーク（ＳＡＮ）は、ストレージ・デバイスをホスト・サーバに接続するネットワークである。このネットワークは、ネットワーキング・インフラストラクチャとしてファイバ・チャネル技術に基づくものである。ＳＡＮと以前の相互接続方式とを区別するものは、ホスト・サーバとストレージとの任意間接続性（any-to-any connectivity）に加えて、集中（簡略）管理を可能にする１つの大規模「ストレージ域」にストレージのすべて（またはほとんどすべて）を統合することができるという基本概念である。

【００１０】

ファイバ・チャネルＳＡＮは、ｚＳｅｒｉｅｓシステムおよびストレージと同じネットワーク内のオープン・システムおよびストレージ（すなわち、非ｚＳｅｒｉｅｓ）の相互接続を可能にする潜在能力を有する。オープン接続とｚＳｅｒｉｅｓ接続の両方のためのプロトコルがファイバ・チャネル・アーキテクチャのＦＣ－４層にマッピングされるので、これはあり得ることである。

【００１１】

ファイバ・チャネル接続では、どのＥＳＳ（ＩＢＭのエンタープライズ・ストレージ・サーバ）ファイバ・チャネル・ポートにホストが接続されるかとは無関係に、ＬＵＮは、ホストのファイバ・チャネル・アダプタへの親和性（affinity）（アダプタのワールド・ワイド固有ＩＤ、別名、ワールド・ワイド・ポート名による）を有する。したがって、単一ファイバ・チャネル・ホストがＥＳＳ上の複数ファイバ・チャネル・ポートにアクセスできるスイッチ・ファブリック構成では、ファイバ・チャネル・ホストによってアクセス可能な複数ＬＵＮのセットはそれぞれのＥＳＳポート上で同じになる。

【００１２】

この実現例の結果の１つは、ファイバ・チャネルでは、ＳＣＳＩとは異なり、ＬＵＮマスキングが各ファイバ・チャネル・ホストごとに異なる可能性があるので、同じファイバ・チャネル・ポートへのファブリックを介してＥＳＳに接続されたホストが同じＬＵＮを「認識する（see）」ことができない可能性があることである。換言すれば、各ＥＳＳは、どのホストがどのＬＵＮにアクセスできるかを定義することができる。

【００１３】

もう１つの方法は、各ホストからの各ファイバ・チャネル・ポートがやむを得ずＥＳＳ上の１つのファイバ・チャネル・ポートに接続するようなゾーンを交換機内に作成し、それにより、ホストが１つのパスのみを介してＬＵＮを認識できるようにすることである。

【００１４】

ファイバ・チャネル仕様の詳細は、「Fibre Channel Physical and Signaling Interface (FC-PH), ANSIX3.230-1994」、「Fibre Channel Second Generation Physical Interface (FC-PH-2),ANSI X3.297-1997」、「Fibre Channel Third Generation Physical Interface (FC-PH-3),ANSI X3.303-199X, Revision 9.4」、および「Fibre Channel Arbitrated Loop (FC-AL), ANSIX3.272-1996」という規格に示されている。他の関連規格としては、FC-FSおよびFC-GS-3がある。

【００１５】

ファイバ・チャネルに関する詳細情報は、「The Fibre Channel Consultant - A Comprehensive Introduction 」（RobertW. Kembel著、１９９８年）および「The Fibre Channel Consultant - Arbitrated Loop 」（Robert W.Kembel著、１９９６年）に開示されている。

【００１６】

２０００年１２月２２日に出願され、２００１年7月１１日に公開され、米国ニューヨーク州アーモンクのインターナショナル・ビジネス・マシーンズ社に譲渡されたBarry Stanley Barnett他によるＥＰ　１　１１５　２２５　Ａ２「Method and system forend-to-end problem determination and fault isolation for storage area networks」では、ストレージ・エリア・ネットワーク（ＳＡＮ）における問題判別および障害分離のための方法およびシステムを開示している。マルチベンダ・ホスト・システム、ＦＣスイッチ、およびストレージ周辺装置からなる複雑な構成は、通信アーキテクチャ（ＣＡ）を介してＳＡＮ内で接続される。通信アーキテクチャ・エレメント（ＣＡＥ）は、ネットワーク・サービス・プロトコルによりホスト・コンピュータ上の通信アーキテクチャ・マネージャ（ＣＡＭ）に正常に登録したネットワーク接続装置であり、ＣＡＭは、ＳＡＮに関する問題判別（ＰＤ）機能を含み、ＳＡＮ　ＰＤ情報テーブル（ＳＰＤＩＴ）を維持する。ＣＡは、ＳＰＤＩＴに保管された情報を通信可能なすべてのネットワーク接続エレメントを有する。ＣＡＭはＳＡＮトポロジ・マップを使用し、ＳＰＤＩＴはＳＡＮ診断テーブル（ＳＤＴ）を作成するために使用される。特定の装置内の障害のあるコンポーネントは、同じネットワーク接続パスに沿った装置にエラーを発生させるようなエラーを発生する可能性がある。ＣＡＭがエラー・パケットまたはエラー・メッセージを受信すると、そのエラーはＳＤＴに保管され、そのエラーとＳＤＴ内の他のエラーとを時間的かつ空間的に比較することにより、各エラーが分析される。そのエラーを発生する候補であるとＣＡＥが判定された場合、そのＣＡＥは可能であれば交換のために報告される。

【００１７】

２００１年２月９日に出願され、２００１年１２月２０日に公開された日本国川崎市のＳａｗａｏ　ＩｗａｔａｎｉによるＵＳ　２００１／００５４０９３　Ａ１「Storage area network management system, method, andcomputer-readable medium」では、単一ソースから伝統的に分散したセキュリティ・システムを統合して管理し、ＳＡＮ内のセキュリティ管理を自動化するストレージ・エリア・ネットワーク（ＳＡＮ）の統合管理メカニズムを開示している。この統合管理メカニズムは、ＳＡＮを統合して管理するものであり、この統合管理メカニズムを使用してＳＡＮのホスト・コンピュータとストレージ・デバイスのアクセス関係が管理されるように構成される。それに対してホスト・コンピュータ

からアクセスが試行されるストレージ・デバイスの一領域を含む統合管理メカニズム上のアクセス・パスと、そのストレージにアクセスするときに使用されるファイバ・チャネル・アダプタと、ホスト・バス・アダプタ（ＨＢＡ）が構成される。構成されたアクセス・パス情報に基づいて、統合管理メカニズムは、それぞれのストレージ設定と、ゾーニング設定と、ホスト・コンピュータのＳＡＮ管理メカニズム、交換機のゾーニング設定メカニズム、およびストレージ・デバイスのストレージ管理メカニズムに関するアクセス可能領域許可を確立する。

【特許文献１】ＥＰ　１　１１５　２２５　Ａ２

【特許文献２】ＵＳ　２００１／００５４０９３　Ａ１

【非特許文献１】「Fibre Channel Physical and Signaling Interface (FC-PH), ANSIX3.230-1994」

【非特許文献２】「Fibre Channel Second Generation Physical Interface (FC-PH-2), ANSIX3.297-1997」

【非特許文献３】「Fibre Channel Third Generation Physical Interface (FC-PH-3), ANSIX3.303-199X, Revision 9.4」

【非特許文献４】「Fibre Channel Arbitrated Loop (FC-AL), ANSI X3.272-1996」

【非特許文献５】FC-FS

【非特許文献６】FC-GS-3

【非特許文献７】「The Fibre Channel Consultant - A Comprehensive Introduction 」（RobertW. Kembel著、１９９８年）

【非特許文献８】「The Fibre Channel Consultant - Arbitrated Loop 」（Robert W. Kembel著、１９９６年）

【発明の開示】

【発明が解決しようとする課題】

【００１８】

ここから始めると、本発明の目的は、改良されたセキュリティ・メカニズムを有し、複数サーバが１つのファイバ・チャネル・アダプタを共用するサーバ環境においてストレージ・エリア・ネットワークを操作するための方法およびシステムを提供することにある。

【課題を解決するための手段】

【００１９】

上記の目的は、独立請求項に述べられている方法およびシステムによって達成される。本発明の他の有利な諸実施形態は、下位請求項に記載されており、以下の説明で教示される。

【００２０】

本発明により、複数サーバが１つのファイバ・チャネル・アダプタを共用するサーバ環境においてストレージ・エリア・ネットワーク（ＳＡＮ）を操作するための方法およびシステムが提供される。ＳＡＮ管理サーバは、ストレージ・システム内の領域およびセキュリティを管理するか、またはエラーを検出してＳＡＮを構成するか、あるいはその両方を行い、ファイバ・チャネル・ネットワークはストレージ・デバイスへの接続を行い、複数のオペレーティング・システム・イメージは前記サーバ環境で実行される。さらに、トラステッド（trusted）ＳＡＮ管理クライアント・ユニットは、前記ＳＡＮ管理サーバおよびファイバ・チャネル・アダプタに接続され、それにより、トラステッドＳＡＮ管理クライアント・ユニットは、前記オペレーティング・システム・イメージのそれぞれの代わりに、前記ファイバ・チャネル・ネットワークにおいてコマンドを発行するように構成される。

【００２１】

本発明の好ましい一実施形態では、サーバ環境は、仮想サーバまたはパーティション・サーバ（partitioned server）あるいはその両方を含む。

【００２２】

好ましくは、ＳＡＮ管理サーバは、第１のコマンド・セットと第２のコマンド・セット

とを区別するように構成され、それにより、第１のコマンド・セットは前記ＳＡＮとともにＳＭクライアントによって処理され、前記第２のコマンド・セットは前記ＳＡＮへのアクセスなしに前記ＯＳイメージによって処理される。都合よく、ファイバ・チャネル・アダプタ（ＦＣアダプタ）は、前記トラステッドＳＡＮ管理クライアント・ユニットを認証するように構成される。

【００２３】

有利には、ＦＣアダプタおよび前記ＳＡＮは、非トラステッド（untrusted）ＯＳイメージのアクセスを最小必要コマンド・セットに制限するように適合させることができる。代わって、ＦＣアダプタおよび仮想サーバの仮想化層は、非トラステッドＯＳイメージのアクセスを最小必要コマンド・セットに制限するように適合させることができる。

【００２４】

本発明の他の実施形態では、サーバ負荷を小さいものに保持するために、１つのＳＭクライアントのみが設けられる。任意選択で、冗長性を提供するために、１つまたは複数のバックアップＳＭクライアントが設けられる。

【００２５】

有利には、ＳＡＮからのメッセージを受信するためにＳＭクライアントのみが登録され、ＳＭクライアントは、前記ＳＭサーバにのみ前記メッセージを転送するように構成される。任意選択で、ＦＣアダプタは、それに関する登録が不要なすべてのメッセージをＳＭクライアントにのみ転送し、非トラステッドＯＳイメージには転送しないように構成される。

【００２６】

本発明の異なる一実施形態では、サーバは、２つのクラスのエージェント、すなわち、ＳＭクライアントおよびリモート・アクセス・サーバ（ＲＡサーバ）を備えている。好ましくは、サーバは、ＲＡサーバにアクセスするための許可データを保持するためのリポジトリを備えている。

【００２７】

好ましくは、ＳＭクライアントおよびＦＣアダプタのみが、各非トラステッドＯＳイメージによるリソースの使用について請求するために使用される情報を収集するように構成される。有利には、ＳＭフレームワークは、アクセス権を設定するためにファイアウォール制御アプリケーションと通信するように適合される。

【００２８】

本発明のさらに他の一実施形態では、ＳＭクライアントは、ＳＭサーバからＲＡサーバへの要求に関するルータとして機能するように適合される。好ましくは、既存のＴｅｌｎｅｔ／ｓｓｈｄサーバがＲＡサーバを形成する。

【００２９】

さらに、本発明は、複数オペレーティング・システム・イメージが１つのファイバ・チャネル・アダプタを共用するサーバ環境においてストレージ・エリア・ネットワーク（ＳＡＮ）を操作するための方法として実現することができ、ＳＡＮは、前記ファイバ・チャネル・アダプタへの通信パスにより少なくとも１つのＳＡＮ管理サーバと少なくとも１つのＳＡＮ管理クライアントとともにＳＡＮ管理ソフトウェアによって管理される。ＳＡＮ管理サーバによって発行された要求は少なくとも２つのグループに分離され、すなわち、第１のグループは、トラステッド・パスに対応して、同じアダプタを共用する他のオペレーティング・システムの代わりにＳＭクライアントを代表してファイバ・チャネル・アダプタおよびＳＡＮによって処理され、第２のグループは、ＦＣアダプタおよびＳＡＮとの間で要求を送受信する必要なしに他のオペレーティング・システムによって処理される。

【００３０】

この方法の好ましい一実施形態では、ＳＡＮおよびＦＣアダプタ内で生成された非送信請求メッセージに含まれるすべての情報がＳＡＮ管理クライアントによってＳＡＮマネージャに経路指定される。好ましくは、ファイアウォールを変更するためにＨＢＡ＿ＡＰＩバインディング要求が使用される。

【００３１】
任意選択で、ＳＡＮ管理クライアントからアダプタへの通信パスは、他のオペレーティング・システム・イメージによって変更または盗聴できないように操作される。さらに他の好ましい一実施形態では、個別オペレーティング・システム・イメージについて請求するためのすべての関連情報がアダプタ内で生成され、ＳＡＮを介してトラステッド・パス上のＳＡＮクライアントによってＳＡＮマネージャに経路指定される。
【００３２】
有利には、ＳＭサーバは、前記第１のグループからの要求を実行するためにＳＭクライアントに許可データを提供する。任意選択で、ＳＭサーバおよびＳＭクライアントは、前記第２のグループからの要求を実行するために他のＯＳイメージに許可データを提供する。好ましくは、ＯＳイメージは、ＳＡＮ内の限定コマンド・セットを実行するためにのみ使用可能になるように操作される。
【００３３】
多数のオペレーティング・システム（＞２５６個、２００４年には４０００台の仮想サーバが考えられる）が、共用ＦＣアダプタとともにＳＡＮに参加するに違いない。このタイプのシナリオでは、コストおよび管理の容易性の理由によりアダプタが共用される。２５６台のＦＣアダプタを備えたサーバは、たとえば、アダプタから交換機への２５６本のケーブルと、交換網内の２５６個のポートを必要とする。
【００３４】
サーバ・ホスティング環境では、各ＯＳイメージは、他のＯＳイメージによってアクセス不能な専用データ（たとえば、サーバ構成）、共用読取り書込みデータ、たとえば、共用データベース、共用読取り専用データ、たとえば、同じアダプタによるＯＳイメージによって共用されるＬＵＮ用のプリインストール済みオペレーティング・システム・イメージ（ＵＮＩＸシステムでは／ｕｓｒ）を必要とし、完全にＳＣＳＩ準拠の動作（たとえば、ＳＡＭ－２によって定義されるように、予約／解放(reserve/release)、ＮＡＣＡ処理、キューイング・ルール）を保証することは可能ではない。
【００３５】
各「非トラステッド・オペレーティング・システム・イメージ」は、潜在的に危険なエンティティ、たとえば、互いに競争する２社またはハッカーによって所有される。所有とは、実際にルート・アクセスでき、すべてのＳＭクライアントまたは他のソフトウェアを含むオペレーティング・システム・イメージのすべての部分を変更できるエンティティを意味する。ＯＳイメージ所有者は、アクセス権をまったく持っておらず、それ自体のためにハードウェアを変更することができないが、マシン所有者がこれを実行することを必要とする。マシン所有者（ＩＴ部門／ＡＳＰ／ＩＳＰ）は、ＳＡＮにおけるすべてのハードウェア、マッピング、およびポリシーに関する完全な制御権を有する。マシン所有者は、ＳＡＮ内のリソース（最も可能性が高いのはディスク・コントローラ内のＬＵＮ）をオペレーティング・システム・イメージに割り当てる。マシン所有者は、オペレーティング・システム・イメージのダウンタイムを防止するためにＳＡＮにおいてエラー検出および障害分離を実行するためのツールを必要とする。マシン所有者は、そのハードウェアを複数エンティティおよびサブネットに分割することを希望する可能性があり、そのサブネットは他のサブネットから独立して管理することができる（複数サーバまたはＬＰＡＲをグループ化(group multiple servers on LPARs)する）。共用アダプタは複数サブネットに及ばないが、複数ＬＰＡＲに及ぶ可能性がある。
【００３６】
各ＯＳイメージは、個別サーバ（ブレード・サーバ）上で実行されるかまたは仮想サーバとして実行される。仮想サーバ環境は、たとえば、ＬＰＡＲ－ｚＳｅｒｉｅｓファームウェア、ｚＳｅｒｉｅｓ　ＶＭオペレーティング・システム、またはＩｎｔｅｌベースのサーバ上のＶＭＷａｒｅによって作成することができる。
【００３７】
ＳＡＮマネージャのユーザ・インターフェースにおけるＳＡＮリソースの提示は、完全

に異なる種類のビューをＳＭサーバのユーザ・インターフェースによって提示させずに、マシン所有者が仮想サーバから物理サーバにＯＳイメージを移動できるように実行しなければならない。

【００３８】

各ＯＳイメージのアクセスは、ＦＣアダプタ内のファイアウォールまたはその他のエンティティによって制限することができ、そのエンティティはＯＳイメージに属さない（さもなければ、ルート・アクセス権を備え、ＯＳイメージを完全に制御できるユーザはそれを回避する能力を有する）。

【００３９】

各非トラステッドＯＳイメージにおけるＦＣからＳＣＳＩへのマッピングは、ＦＣデバイス・ドライバ・マッピング構成インターフェースＭＡＰ＿ＩＦによって構成することができる。たとえば、ｚＳｅｒｉｅｓ上の特定のバージョンのファイバ・チャネル上のＭＡＰ＿ＩＦは、前記マッピングを設定し照会するためのｐｒｏｃ－ｆｉｌｅ－ｓｙｓｔｅｍというＬｉｎｕｘ特定構成メソッドによって実現される。ＳＭサーバは、ＭＡＰ＿ＩＦによって報告される正しいデータを当てにしてはならない（アクセス強制（access enforcement）はファイアウォールによって実行され、したがって、ＯＳは協働するかまたはいかなるデータもまったく認識しなくなる）。請求のための測定は非トラステッドＯＳイメージ内で実行することができない（ルート・ユーザが自由にデータを変更できるからである）。したがって、アダプタまたはその他のエンティティは、必要な測定データを提供しなければならない。

【００４０】

他の諸実施形態は複数のＷＷＰＮを提供することができるが、ＷＷＰＮの数はアダプタごとにサポートされるＯＳイメージの数より小さくなる可能性がある。

【００４１】

本発明の上記ならびに追加の目的、特徴、および利点は、以下に詳細に記載した説明で明らかになるであろう。

【００４２】

本発明の新規の特徴は特許請求の範囲に示されている。しかし、本発明そのもの、ならびにその好ましい使用態様、その他の目的、および利点は、添付図面に併せて読んだときに例示的な一実施形態に関する以下の詳細な説明を参照することにより、最も良く理解されるであろう。

【発明を実施するための最良の形態】

【００４３】

図１に関しては、本発明によるシステム１００の第１の実施形態を示すブロック図が示されている。システム１００は複数のコンピュータ・システム１０４、１０５、および１０６を有し、いずれもＷＷＰＮ１（ワールド・ワイド・ポート名）を有する１つの共通ファイバ・チャネル・アダプタ１１２を介してＳＡＮ１１０にアクセスするように適合されている。明瞭にするために、３台のコンピュータ・システムのみが示されている。コンピュータ・システムの数は百の位または千の位にもすることができることが認知されている。このような多数のコンピュータ・システムを有するので、システム内には２つ以上のファイバ・チャネル・アダプタが存在する可能性があるが、このケースでは複数のコンピュータ・システムが１つかつ同じファイバ・チャネル・アダプタを共用することになるであろう。

【００４４】

それぞれのコンピュータ・システム１０４、１０５、および１０６がＳＡＮ１１０の許可部分にアクセスするためにのみ使用可能になることを保証するために、それぞれのコンピュータ・システムとファイバ・チャネル・アダプタとの間のゲートウェイとして、ファイアウォール１１４、１１５、および１１６がそれぞれ設けられている。ファイアウォール１１４、１１５、および１１６は、ファイバ・チャネル・アダプタに統合するかまたはそれに接続することができる。これに反して、すべてのコンピュータ・システム１０４、

１０５、および１０６は、イーサネット・ネットワークなどのネットワーク１２０に接続される。

【００４５】

さらに、システム１００は、ネットワーク１２０に接続され、オペレーティング・システム（ＯＳ）１３１上で実行されるＳＡＮ管理サーバ１３０（ＳＭサーバ）およびＳＡＮ管理クライアント１３２（ＳＭクライアント）をそれぞれホストする、さらに2台のコンピュータ・システム１２２および１２３を有する。ＳＭサーバ１３０はＳＡＮ管理システム、たとえば、Ｔｉｖｏｌｉ Ｓｔｏｒａｇｅ Ｎｅｔｗｏｒｋ Ｍａｎａｇｅｒ（http://www.tivoli.com/products/index/storage_net_mgr/）のサーバ部分を実行し、ＳＭクライアント１３２はこのようなシステムのそれぞれのクライアント部分を実行する。

【００４６】

ＳＭサーバ１３０は通信回線を介してファイアウォール制御アプリケーション１３４にさらに接続され、ＳＭクライアント１３２はファイバ・チャネル・アダプタ１１２への通信リンクをすでに取得している。ファイアウォール制御アプリケーション１３４には、ファイバ・チャネル・アダプタ（図示の通り）によるかまたはそれぞれに対して直接的に、ファイアウォール１１４、１１５、および１１６のそれぞれへの通信リンクが設けられている。

【００４７】

ＳＭサーバ１３０は、それぞれのセキュア・シェル・インターフェース１４４、１４５、および１４６、すなわち、リモート・コンピュータにログインし、リモート・コンピュータ上でコマンドを実行するためのプログラムを介してコンピュータ・システム１０４、１０５、および１０６のそれぞれにアクセスする。各コンピュータ・システム１０４、１０５、および１０６内に設けられたマッピング・インターフェースＭＡＰ＿ＩＦ１５４、１５５、１５６は、ファイバ・チャネルからＳＣＳＩへのマッピングを処理する。これは、たとえば、コマンド行インターフェースまたは構成ファイルとして実現することができる。

【００４８】

各コンピュータ・システム１０４、１０５、および１０６は、「非トラステッド」である可能性があるオペレーティング・システム１６４、１６５、および１６６を実行する。これに関連して「非トラステッド」とは、潜在的に危険なエンティティ、たとえば、コンピュータ・ウィルスなどの１つの悪意のあるコードまたは前記エンティティにとって到達不能でなければならないＳＡＮ内の情報にアクセスするかまたはそれを変更するためにオペレーティング・システムを改ざんしようとする人によってオペレーティング・システムが制御または操作される可能性があることを意味する。オペレーティング・システム自体は、インターナショナル・ビジネス・マシーンズ社によるＡＩＸまたはｚ／ＯＳ、ＵＮＩＸ、およびＬｉｎｕｘなどの多種多様なオペレーティング・システムのいずれかによって形成することができる。

【００４９】

本発明によるセキュリティ・メカニズムは、コンピュータ・システム１０４、１０５、および１０６のうちのいずれか１つによって無許可ＳＡＮアクセス要求が発行される可能性があることを考慮するので、それぞれのファイアウォール１１４、１１５、および１１６はすべてのＳＡＮアクセス要求をフィルタリングし、ファイアウォール制御アプリケーションによって許可されたもののみを受け入れる。許可ＳＡＮアクセス要求を指定するファイアウォール設定を変更するために、コンピュータ・システム１０４、１０５、および１０６のいずれでもアクセス不能なファイアウォール制御アプリケーション１３４のみが使用可能になる。本発明の第１の実施形態によれば、ＳＭサーバ１３０はファイアウォール制御アプリケーション１３４を制御する。

【００５０】

ＳＭクライアント１３２は、「トラステッド」でなければならないオペレーティング・システム１７０の上でＨＢＡ＿ＡＰＩ１６８（ホスト・バス・アダプタ・アプリケーショ

ン・プログラム・インターフェース）の上で実行され、すなわち、オペレーティング・システムはＳＡＮアクセス権などを改ざんする意図がないエンティティによって管理される。ファイバ・チャネル・アダプタ１１２は、非トラステッド・オペレーティング・システムとトラステッド・オペレーティング・システムとを区別するように適合される。この許可は、本発明の一部ではない固定ソース・アドレス（たとえば、ハードウェアＩＤ）、キーおよび暗号アルゴリズム、またはパスワードのような周知の許可方式によりオペレーティング・システム１７０またはＳＭクライアント１３２を識別することによって実施することができる。ＳＭクライアント１３２を実行するトラステッド・オペレーティング・システム１７０からの、ＳＡＮ構成、ＳＡＮコンポーネント、ＳＡＮアクセス権、エラー・メッセージ、統計または請求情報などの重大な情報に関連する要求のみを受け入れることができる。これに対応して、このような要求に対する結果は、トラステッド・オペレーティング・システム１７０にのみ返される。換言すれば、ＳＭサーバによって制御され、非トラステッド・オペレーティング・システムの代わりに動作するＳＭクライアント１３２のみがＳＡＮ管理を実行する。

【００５１】

ＳＭサーバ１３０は、ＳＡＮおよび管理対象オペレーティング・システムで情報を照会し設定するために要求を発行するＳＭコア・エンジン（図示せず）と、その要求および応答をコア・エンジンからクライアントに経路指定する通信モジュール（図示せず）とを有する。通信モジュールは、ＳＭサーバの一部として実現するか、または「トラステッド」環境に存在するＳＭサーバおよびＳＭクライアント・コンポーネントに分散することができる。

【００５２】

セキュア・シェル・デーモンｓｓｈｄおよびＯＳ特定ＦＣ構成インターフェースはＲＡ（リモート・アクセス）サーバとして要約される。リモート・アクセス・サーバは、ＳＭサーバまたはＳＭクライアントによって送信された許可要求に応答するように適合される。好ましい実現例では、ＳＭクライアントおよびＲＡサーバは、要求の発信元を識別するために、パスワード、ユーザＩＤ、および暗号キーなどの許可データを使用する。したがって、通信モジュールは、前記許可情報に関するリポジトリ（図示せず）を備えている。通信モジュールは、以下のコマンド・リストに応じて、ファブリック関連要求、すなわち、ＳＡＮを管理するための要求と、アダプタ関連要求、すなわち、ファイバ・チャネル・アダプタを管理するための要求をＯＳ特定要求から分離する。前述の通り、ＳＭサーバと、ＳＭクライアントと、ＲＡサーバとの間の接続は、たとえば、イーサネット上で動作するＩＰベースのネットワークである。異なる一実施形態では、ＳＭクライアントと、ＳＭサーバと、ＲＡサーバとの間の何らかの通信は、ＦＣアダプタの諸機能を使用して、個別ネットワークの代わりに他のプロトコルをトランスポートすることができるであろう。一例は、ファイバ・チャネルによるＴＣＰ／ＩＰになるであろう。

【００５３】

次に図２に関しては、本発明によるシステムの第２の実施形態を示すブロック図が示されている。

【００５４】

第１の実施形態（図１）のシステムに対応して、このシステムは複数のコンピュータ・システム２０４、２０５、および２０６を有し、いずれもＷＷＰＮ１（ワールド・ワイド・ポート名）を有する１つの共通ファイバ・チャネル・アダプタ２１２を介してＳＡＮ２１０にアクセスするように適合されている。明瞭にするために、３台のコンピュータ・システムのみが示されている。コンピュータ・システムの数は百の位または千の位にもすることができることが認知されている。このような多数のコンピュータ・システムを有するので、システム内には２つ以上のファイバ・チャネル・アダプタが存在する可能性があるが、このケースでは複数のコンピュータ・システムが１つかつ同じファイバ・チャネル・アダプタを共用することになるであろう。

【００５５】

それぞれのコンピュータ・システム２０４、２０５、および２０６がＳＡＮ２１０の許可部分にアクセスするためにのみ使用可能になることを保証するために、それぞれのコンピュータ・システムとファイバ・チャネル・アダプタとの間のゲートウェイとして、ファイアウォール２１４、２１５、および２１６がそれぞれ設けられている。ファイアウォール２１４、２１５、および２１６は、ファイバ・チャネル・アダプタに統合するかまたはそれに接続することができる。

【００５６】

第１の実施形態とは反対に、ＬＰＡＲ（ロジカル・パーティション・モード）およびＶＭ（仮想計算機）などの仮想サーバ環境で動作する仮想サーバがコンピュータ・システム２０４、２０５、および２０６を形成する。さらに、ＳＡＮ管理クライアント２３２（ＳＭクライアント）をホストするために、もう１つの仮想サーバ２２３が設けられている。コンピュータ・システム２０４、２０５、および２０６は、ハイパーソケットの上のセキュア・シェル接続を介してＳＭクライアント２３２と通信する。

【００５７】

ＳＡＮ管理サーバ２３０（ＳＭサーバ）をホストするために個別コンピュータ・システム２２２が設けられている。ＳＭサーバ２３０はＳＡＮ管理システム、たとえば、Ｔｉｖｏｌｉ Ｓｔｏｒａｇｅ Ｎｅｔｗｏｒｋ Ｍａｎａｇｅｒのサーバ部分を実行し、ＳＭクライアント２３２はこのようなシステムのそれぞれのクライアント部分を実行する。ＳＭサーバ２３０はイーサネット・ネットワーク２２０を介してＳＭクライアントにアクセスする。

【００５８】

各コンピュータ・システム２０４、２０５、および２０６は、「非トラステッド」である可能性がある（上記を参照）オペレーティング・システム２６４、２６５、および２６６を実行する。オペレーティング・システム自体は、インターナショナル・ビジネス・マシーンズ社によるＡＩＸまたはｚ／ＯＳ、ＵＮＩＸ、およびＬｉｎｕｘなどの多種多様なオペレーティング・システムのいずれかによって形成することができる。

【００５９】

ＳＭクライアント２３２は、「トラステッド」でなければならい（上記を参照）オペレーティング・システム２７０の上でＨＢＡ＿ＡＰＩ２６８（ホスト・バス・アダプタ・アプリケーション・プログラム・インターフェース）の上で実行される。この場合も、ファイバ・チャネル・アダプタ２１２は、非トラステッド・オペレーティング・システムとトラステッド・オペレーティング・システムとを区別するように適合される。ＳＭクライアント２３２を実行するトラステッド・オペレーティング・システム２７０からの重大な情報に関連する要求のみを受け入れることができる。ファイアウォール制御アプリケーション２３４も「トラステッド」オペレーティング・システム２７０の上で実行される。ファイアウォール制御アプリケーションは、特定のアクセス要求を「非トラステッド」オペレーティング・システム２６４、２６５、および２６６からＳＡＮ２１０に転送するかどうかをファイアウォール２１４、２１５、および２１６に指示するために使用される。換言すれば、許可ＳＡＮアクセス要求を指定するファイアウォール設定を変更するために、コンピュータ・システム２０４、２０５、および２０６のいずれでもアクセス不能なファイアウォール制御アプリケーション２３４のみが使用可能になる。

【００６０】

次に図３に関しては、本発明により複数サーバが１つのファイバ・チャネル・アダプタを共用するサーバ環境においてストレージ・エリア・ネットワークを操作するための方法を示す流れ図（ブロック３００）が示されている。

【００６１】

第一に、ＳＭコア・エンジンは要求を作成し（ブロック３０２）、次に通信モジュールは要求のターゲットを決定する（ブロック３０６）。それがＳＡＮまたはファイバ・チャネル・アダプタに関する要求である場合、通信モジュールはそれぞれの許可データによりＳＭクライアントへの通信パスを確立し（ブロック３０８）、許可データはパスワード、

ユーザＩＤ、および暗号キーにすることができる。

【００６２】

その後、ＳＭクライアントは、その許可が有効であるかどうかをチェックする（ブロック３１２）。ｙｅｓである場合、ＳＭクライアントは、交換機（たとえば、ＳＡＮ内のＦＣ装置のリストを検索するため）またはディスク・コントローラ（たとえば、コントローラ内の論理ディスクのリストを検索するため）などのＳＡＮ内のエンティティに照会し、特定のタイプのエラー通知メッセージについて登録するために使用されるＲＮＩＤ－ＥＬＳなどのファイバ・チャネル・アダプタおよびＳＡＮの属性を設定することによって応答を作成する（ブロック３１４）。

【００６３】

ｎｏである場合、ＳＭクライアントは、その要求が拒否されたことをＳＭコアに通知するリジェクト応答を作成する（ブロック３１６）。両方の代替パスは、通信システムに応答を送信することによって継続する（ブロック３１８）。

【００６４】

ブロック３０６に戻り、HBAGetFcpTargetMappingFunc関数によって定義されたファイバ・チャネルからＳＣＳＩへのマッピングなどのオペレーティング・システム（ＯＳ）構成データに関する要求をＳＭコア・エンジンが作成したと通信モジュールが判定した場合、通信モジュールはそれぞれの許可データによりＲＡサーバへの通信パスを確立する（ブロック３２０）。

【００６５】

次に、ＲＡサーバ内の許可コンポーネント、たとえば、ｓｓｈｄは、提示されたユーザＩＤ、パスワード、およびキーと、許可すべきＲＡサーバが把握しているユーザＩＤ、パスワード、およびキーとを比較することにより、その許可が有効であるかどうかを判定する（ブロック３２１）。

【００６６】

ｙｅｓである場合、ＲＡサーバは、オペレーティング・システムのファイバ・チャネル・デバイス・ドライバによって保管されたファイバ・チャネル構成情報にアクセスすることなどのＯＳ構成操作によって応答を作成する（ブロック３２４）。

【００６７】

ｎｏである場合、ＲＡサーバはリジェクト応答を作成する（ブロック３２６）。この場合も、両方の代替パスは、通信システムに応答を送信することによって継続する（ブロック３１８）。その後、通信システムはＳＭコア・エンジンに応答を転送し（ブロック３２０）、ＳＭサーバ内のＳＭコア・エンジンは、言及した２件の特許に定義されている通り、応答を処理する（ブロック３２２）。

【００６８】

以下では、好ましい実現例の要求のフローおよびタイプが示されている。ＨＢＡ＿ＡＰＩコマンドの構文は「Fibre Channel HBA API Working draft」（ftp://ftp.t11.org/t11/pub/fc/hba/02-268v2.pdf）で見つけることができる。

【００６９】

１．第一に、ファブリックおよびアダプタ関連要求のＣＴ、ＥＬＳ、ＳＣＳＩコマンドがリストされる。アダプタおよびＳＡＮリソースを使用するＳＭクライアントはこれらのコマンドを処理する。

【数１】

```
typedef HBA_STATUS(* HBASendCTPassThruFunc) (HBA_HANDLE, void *,
HBA_UINT32, void *, HBA_UINT32);
typedef HBA_STATUS (* HBASendRNIDFunc)(HBA_HANDLE, HBA_WWN,
HBA_WWNTYPE, void *,HBA_UINT32 *);
typedef HBA_STATUS (* HBASendScsiInquiryFunc)(HBA_HANDLE,
HBA_WWN,HBA_UINT64, HBA_UINT8,HBA_UINT32, void *, HBA_UINT32,
void *, HBA_UINT32);
typedef HBA_STATUS (* HBASendReportLUNsFunc)(HBA_HANDLE, HBA_WWN
void *, HBA_UINT32, void *,HBA_UINT32);
typedef HBA_STATUS (* HBASendReadCapacityFunc)(HBA_HANDLE,
HBA_WWN, HBA_UINT64, void *, HBA_UINT32,void *, HBA_UINT32);
typedef HBA_STATUS (* HBASendCTPassThruV2Func)(HBA_HANDLE,
HBA_WWN, void *, HBA_UINT32, void *,HBA_UINT32 *);
typedef HBA_STATUS (* HBASendRNIDV2Func) (HBA_HANDLE, HBA_WWN,
HBA_WWN, HBA_UINT32, HBA_UINT32, void *,HBA_UINT32*);
typedef HBA_STATUS (* HBAScsiInquiryV2Func)
(HBA_HANDLE,HBA_WWN,HBA_WWN, HBA_UINT64, HBA_UINT8, HBA_UINT8,
void *, HBA_UINT32 *, HBA_UINT8 *,void *, HBA_UINT32 *);
typedef HBA_STATUS (* HBAScsiReportLUNsV2Func)(HBA_HANDLE,
HBA_WWN, HBA_WWN, void *, HBA_UINT32 *,HBA_UINT8 *, void *,
HBA_UINT32 *);
typedef HBA_STATUS (* HBAScsiReadCapacityV2Func)(HBA_HANDLE,
HBA_WWN, HBA_WWN, HBA_UINT64, void *,HBA_UINT32 *, HBA_UINT8 *,
void *, HBA_UINT32 *);
typedef HBA_STATUS (* HBASendRPLFunc)(HBA_HANDLE, HBA_WWN,
HBA_WWN, HBA_UINT32,HBA_UINT32, void *, HBA_UINT32 *);
typedef HBA_STATUS (* HBASendRPSFunc)(HBA_HANDLE, HBA_WWN,
HBA_WWN, HBA_UINT32, HBA_WWN,HBA_UINT32, void *, HBA_UINT32 *);
typedef HBA_STATUS (* HBASendSRLFunc)(HBA_HANDLE, HBA_WWN,
HBA_WWN, HBA_UINT32, void *,HBA_UINT32 *);
typedef HBA_STATUS (* HBASendLIRRFunc)(HBA_HANDLE, HBA_WWN,
HBA_WWN, HBA_UINT8, HBA_UINT8,void *, HBA_UINT32 *);
```

【数２】

```
typedef HBA_HANDLE (* HBAOpenAdapterFunc)(char *);
typedef void (* HBACloseAdapterFunc)(HBA_HANDLE);
typedef HBA_STATUS (* HBAGetAdapterAttributesFunc)(HBA_HANDLE,
HBA_ADAPTERATTRIBUTES *);
typedef HBA_STATUS (* HBAGetAdapterPortAttributesFunc)
(HBA_HANDLE, HBA_UINT32, HBA_PORTATTRIBUTES *);
typedef HBA_STATUS (* HBAGetPortStatisticsFunc) (HBA_HANDLE,
HBA_UINT32, HBA_PORTSTATISTICS *);
typedef HBA_STATUS (* HBAGetDiscoveredPortAttributesFunc)
(HBA_HANDLE, HBA_UINT32, HBA_UINT32,HBA_PORTATTRIBUTES *);
typedef HBA_STATUS (* HBAGetPortAttributesByWWNFunc) (HBA_HANDLE,
HBA_WWN, HBA_PORTATTRIBUTES *);
typedef void (* HBARefreshInformationFunc)(HBA_HANDLE);
typedef void (* HBAResetStatisticsFunc)(HBA_HANDLE, HBA_UINT32);
typedef HBA_STATUS (* HBAGetEventBufferFunc) (HBA_HANDLE,
HBA_EVENTINFO *, HBA_UINT32 *);
typedef HBA_STATUS (* HBASetRNIDMgmtInfoFunc) (HBA_HANDLE,
HBA_MGMTINFO *);
typedef HBA_STATUS (* HBAGetRNIDMgmtInfoFunc) (HBA_HANDLE,
HBA_MGMTINFO *);
typedef HBA_STATUS (* HBAOpenAdapterByWWNFunc) (HBA_HANDLE *,
HBA_WWN);
typedef void (* HBARefreshAdapterConfigurationFunc) ();
typedef HBA_UINT32 (*
HBAGetVendorLibraryAttributesFunc)(HBA_LIBRARYATTRIBUTES *);
typedef HBA_STATUS (* HBAGetFC4StatisticsFunc)(HBA_HANDLE,
HBA_WWN, HBA_UINT8,HBA_FC4STATISTICS *);
typedef HBA_STATUS (* HBAGetFCPStatisticsFunc)(HBA_HANDLE, const
HBA_SCSIID *,HBA_FC4STATIS
typedef HBA_UINT32 (* HBAGetNumberOfAdaptersFunc)();
typedef HBA_STATUS (* HBAGetAdapterNameFunc)(HBA_UINT32, char
*);
```

【００７０】

図４は、前述のタイプのＨＢＡ＿ＡＰＩコマンドに関する簡略信号およびフローを示す流れ図を示している。図４から明らかなように、ＳＭサーバはＳＭクライアントと通信し、そのＳＭクライアントはＦＣスイッチと通信する。ＳＭサーバ、ＳＭクライアント、およびＦＣスイッチは、ブロック４０２、４０４、および４０６によって示す通り、始動さ

れ、互いに独立して動作する。以下の諸ステップは操作の１つのセグメントを構成するだけであり、以下に記載する通り、この方法の前後により多くの要求が送信または受信されることが認知されている。

【００７１】

初めに、ＳＭサーバはＳＭクライアントに要求を送信する（ブロック４０８、４１０）。この例では、それはHBASendLIRRFuncである。詳細には、ＳＭクライアントは、ファイバ・チャネル規格ＦＣ－ＦＳおよびＦＣ－ＧＳ３によって定義された通り、ＣＴ＿ＩＵ、ＥＬＳ、またはＦＣＰ＿ＣＭＤシーケンスをファブリックに送信するために、ＨＢＡ　ＡＰＩを使用する（４１２、４１４）。

【００７２】

ＦＣ規格によって定義された通り、ファブリックで生成された応答は、ＨＢＡ＿ＡＰＩコールの完了部分によってＳＭクライアントに転送される（４１８、４２０）。ＳＭクライアントは前記応答をＳＭサーバに転送する（４２２、４２４）。

【００７３】

２．ＦＣＰ←→ＳＣＳＩマッピング・コマンド

これらのコマンドは、ＯＳおよびＯＳデバイス・ドライバ・リソース・コマンドを使用するＲＡサーバを介して処理される。構文は「Fibre Channel HBA API Working draft」（ftp://ftp.t11.org/t11/pub/fc/hba/02-268v2.pdf）で見つけることができる。

【数３】

```
typedef HBA_STATUS (* HBAGetFcpTargetMappingFunc)(HBA_HANDLE,
HBA_FCPTARGETMAPPING *);
typedef HBA_STATUS (*
HBAGetFcpPersistentBindingFunc)(HBA_HANDLE, HBA_FCPBINDING *);
typedef HBA_STATUS (* HBAGetBindingCapabilityFunc)(HBA_HANDLE,
HBA_WWN, HBA_BIND_CAPABILITY *);
typedef HBA_STATUS (* HBAGetBindingSupportFunc) (HBA_HANDLE,
HBA_WWN, HBA_BIND_CAPABILITY *);
typedef HBA_STATUS (* HBASetBindingSupportFunc) (HBA_HANDLE,
HBA_WWN, HBA_BIND_CAPABILITY);
typedef HBA_STATUS (* HBASetPersistentBindingV2Func) (HBA_HANDLE,
HBA_WWN, const HBA_FCPBINDING2 *);
typedef HBA_STATUS (* HBAGetPersistentBindingV2Func) (HBA_HANDLE,
HBA_WWN, HBA_FCPBINDING2 *);
typedef HBA_STATUS (* HBARemovePersistentBindingFunc)
(HBA_HANDLE, HBA_WWN, const HBA_FCPBINDING2 *);
typedef HBA_STATUS (*
HBARemoveAllPersistentBindingsFunc)(HBA_HANDLE, HBA_WWN);
typedef HBA_STATUS (* HBAGetFcpTargetMappingV2Func)(HBA_HANDLE,
HBA_WWN, HBA_FCPTARGETMAPPING *);
```

【００７４】

次に図５に関しては、前述のタイプのＨＢＡ＿ＡＰＩコマンドに関する簡略信号およびフローを示す流れ図が示されている。図５から明らかなように、ＳＭサーバはこの場合も

ＳＭクライアントと通信する。ＳＭクライアントは返報としてファイアウォールおよびＯＳイメージとそれぞれ通信する。ＳＭサーバ、ＳＭクライアント、ファイアウォール、およびＯＳイメージは、ブロック５０２、５０４、５０６、および５０８によって示す通り、始動され、互いに独立して動作する。以下の諸ステップは操作の１つのセグメントを構成するだけであり、以下に記載する通り、この方法の前後により多くの要求が送信または受信されることが認知されている。

【００７５】

第一に、ＳＭサーバは、要求、たとえば、HBASetPersistentBindingV2というＨＢＡ＿ＡＰＩ要求に対応する要求をＳＭクライアントに送信する（ブロック５１０、５１２）。ＳＭクライアントがファイアウォールを直接制御する場合、これがファイアウォール・セキュリティ・ポリシーによって要求されるのであれば、任意選択でファイアウォールを変更することができる（５１４、５１６、５１８、５２０）。次にＳＭクライアントは、独自のファイアウォール更新メッセージの送信操作をトリガすることにより、ファイアウォールを変更する（５１４、５１６）。

【００７６】

ファイアウォールは、ＳＭクライアントに操作の完了をシグナル通知する（５１８、５２０）。次にＳＭクライアントは、ＲＡサーバにより非トラステッドＯＳイメージ（５２６）において設定または照会要求をトリガする（５２２、５２４）。ＳＭクライアントは前記要求の完了メッセージを待ち（５２８、５３０）、ＳＭクライアントは前記要求の応答をＳＭサーバに返す（５３２、５３４）。

【００７７】

３．着信ＥＬＳ（ＲＮＩＤ）

これらは、ファブリックで開始され、ＳＭサーバに転送する必要があるメッセージである。これらのメッセージは、ＳＡＮで発生する問題の発生源を識別するために使用される。

【００７８】

着信ＥＬＳを処理するためにＨＢＡ＿ＡＰＩで定義されたコマンド：

【数４】

```
typedef HBA_STATUS (* HBARegisterForAdapterAddEventsFunc)(void
(*)(void *, HBA_WWN, HBA_UINT32),void *,HBA_CALLBACKHANDLE *);
typedef HBA_STATUS (* HBARegisterForAdapterEventsFunc)(void
(*)(void *, HBA_WWN, HBA_UINT32),void *,HBA_HANDLE,
HBA_CALLBACKHANDLE *);
typedef HBA_STATUS (* HBARegisterForAdapterPortEventsFunc)(void
(*)(void *, HBA_WWN, HBA_UINT32, HBA_UINT32),void *, HBA_HANDLE,
HBA_WWN, HBA_CALLBACKHANDLE *);
typedef HBA_STATUS (* HBARegisterForLinkEventsFunc)(void
(*)(void *, HBA_WWN, HBA_UINT32, void *,HBA_UINT32),void *, void
*, HBA_UINT32, HBA_HANDLE,HBA_CALLBACKHANDLE *);
typedef HBA_STATUS (*
HBARegisterForAdapterPortStatEventsFunc)(void (*)(void *,
HBA_WWN, HBA_UINT32), void *,HBA_HANDLE, HBA_WWN,
HBA_PORTSTATISTICS,HBA_UINT32, HBA_CALLBACKHANDLE *);
typedef HBA_STATUS (* HBARegisterForTargetEventsFunc)(void
(*)(void *, HBA_WWN, HBA_WWN, HBA_UINT32),void *, HBA_HANDLE,
HBA_WWN, HBA_WWN,HBA_CALLBACKHANDLE *, HBA_UINT32 );
typedef HBA_STATUS (*
HBARemoveCallbackFunc)(HBA_CALLBACKHANDLE);
```

【００７９】

次に、図６に関しては、前述のタイプのＨＢＡ＿ＡＰＩコマンドに関する簡略信号およびフローを示す流れ図を示している。図６から明らかなように、ＳＭサーバはＳＭクライアントと通信し、そのＳＭクライアントはＦＣスイッチと通信する。ＳＭサーバ、ＳＭクライアント、およびＦＣスイッチは、ブロック６０２、６０４、および６０６によって示す通り、始動され、互いに独立して動作する。以下の諸ステップは操作の１つのセグメントを構成するだけであり、以下に記載する通り、この方法の前後により多くの要求が送信または受信されることが認知されている。

【００８０】

ＳＭサーバ（６０２）は、ＨＢＡ＿ＡＰＩによるイベントの場合に１回登録するようＳＭクライアント（６０４）に指示し（６０８、６１０）、完了の確認を待ち（６１２、６１４）、非トラステッドＯＳイメージは登録が許可されない。これが行われた後、ＳＡＮによって作成された各メッセージ（６１６）が以下の手順をトリガする。

１．ＳＭクライアントはＦＣアダプタからイベントを受信する（６１６、６１８）。

２．ＳＭクライアントはＳＭサーバにイベントを転送する（６２０、６２２）。

【００８１】

代替実現例では、ＳＭクライアントはメッセージをフィルタリングし圧縮して（６１６、６１８）、ＳＭサーバに送信されるメッセージの数を削減することができる。

【００８２】

本発明に関連しないＨＢＡ＿ＡＰＩ関数：

【数５】

```
HBA_API functions which are not relevant to this invention:
typedef HBA_UINT32 (* HBAGetVersionFunc)();
typedef HBA_STATUS (* HBALoadLibraryFunc)();
typedef HBA_STATUS (* HBAFreeLibraryFunc)();
```

【００８３】

本発明は、ハードウェア、ソフトウェア、またはハードウェアとソフトウェアの組合せで実現することができる。どのような種類のコンピュータ・システムも、または本明細書に記載した方法を実行するために適合されたその他の装置も適している。ハードウェアとソフトウェアの典型的な組合せは、ロードされ実行されたときに、本明細書に記載した方法を実行するようにコンピュータ・システムを制御するコンピュータ・プログラムを備えた汎用コンピュータ・システムにすることができるであろう。また、本発明は、本明細書に記載した方法の実現例を使用可能にするすべての特徴を有し、コンピュータ・システムにロードされたときにその方法を実行することができるコンピュータ・プログラムに組み込むこともできる。

【００８４】

これに関連して、コンピュータ・プログラム手段またはコンピュータ・プログラムとは、直接的にあるいはａ）他の言語、コード、または表記への変換またはｂ）異なる物質的形態での複製のいずれか一方または両方の後で、情報処理機能を有するシステムに特定の機能を実行させるための１組の命令を任意の言語、コード、または表記で表した任意の表現を意味する。

【図面の簡単な説明】

【００８５】

【図１】本発明によるシステムの第１の実施形態を示すブロック図である。

【図２】本発明によるシステムの第２の実施形態を示すブロック図である。

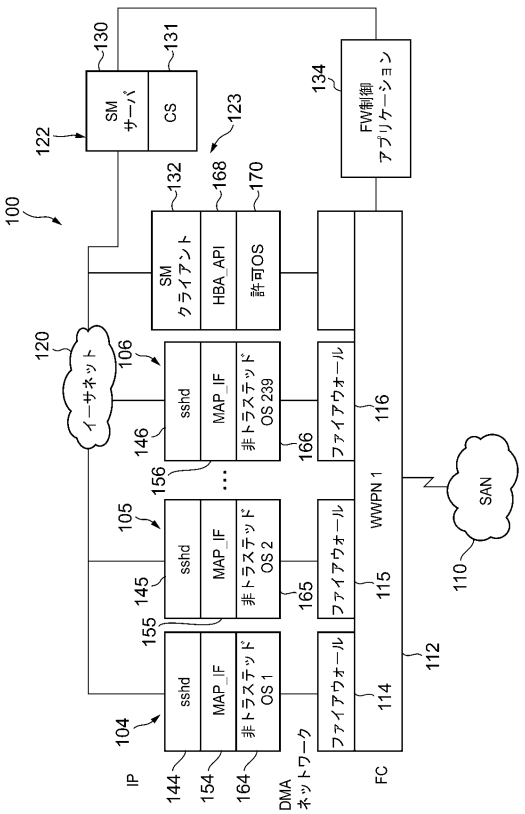
【図３】本発明により複数サーバが１つのファイバ・チャネル・アダプタを共用するサーバ環境においてストレージ・エリア・ネットワークを操作するための方法を示す流れ図である。

【図４】前述のタイプのファブリック関連ＨＢＡ＿ＡＰＩコマンドに関する簡略信号およびフローを示す流れ図である。

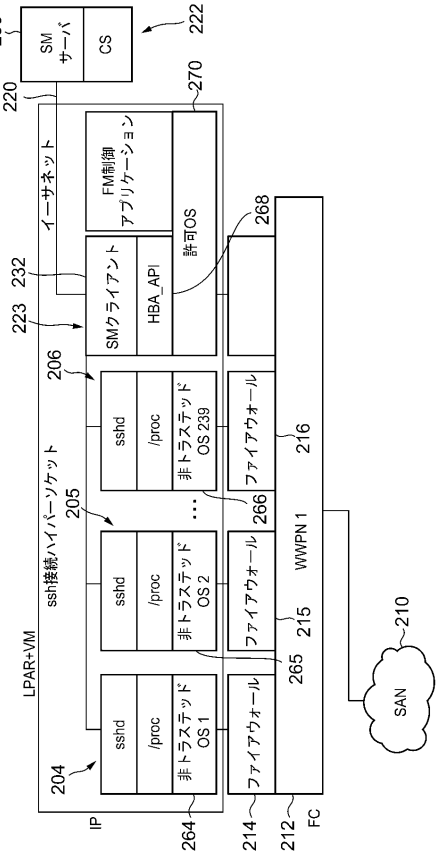
【図５】前述のタイプのＦＣＰからＳＣＳＩへのマッピングのＨＢＡ＿ＡＰＩコマンドに関する簡略信号およびフローを示す流れ図である。

【図６】ＳＡＮによって開始されたＥＬＳ要求を処理するための前述のタイプのＨＢＡ＿ＡＰＩコマンドに関する簡略信号およびフローを示す流れ図である。

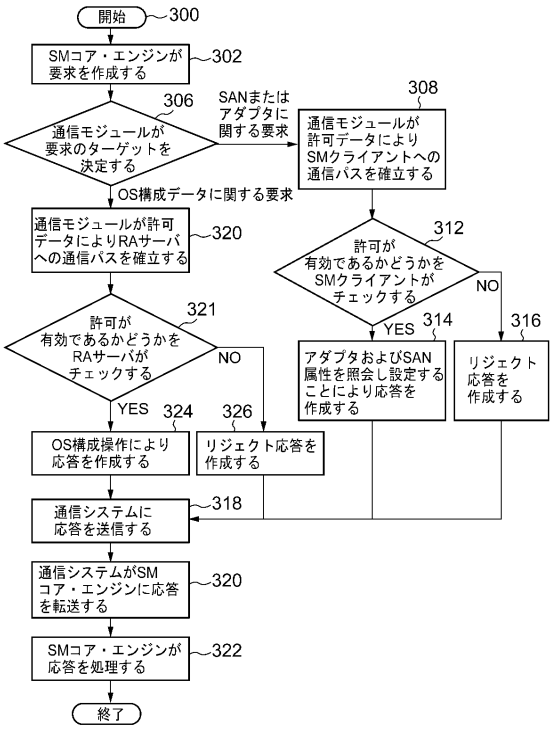
【図１】


SM サーバ
CS
SM クライアント
HBA_API
許可OS
FW制御 アプリケーション
イーサネット
sshd
MAP_IF
非トラステッド OS 239
非トラステッド OS 2
非トラステッド OS 1
ファイアウォール
WWPN 1
SAN
IP
DMA ネットワーク
FC

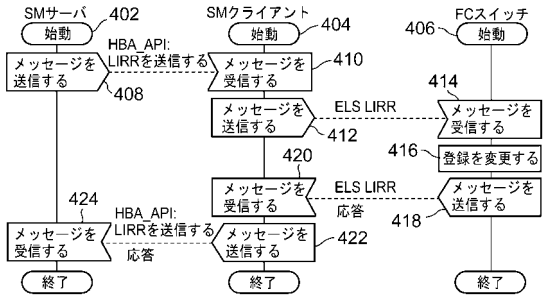
【図２】


SM サーバ
CS
FW制御 アプリケーション
許可OS
イーサネット
SMクライアント
HBA_API
ssh接続ハイパーソケット
sshd
/proc
非トラステッド OS 239
非トラステッド OS 2
非トラステッド OS 1
ファイアウォール
WWPN 1
SAN
LPAR+VM
IP
FC

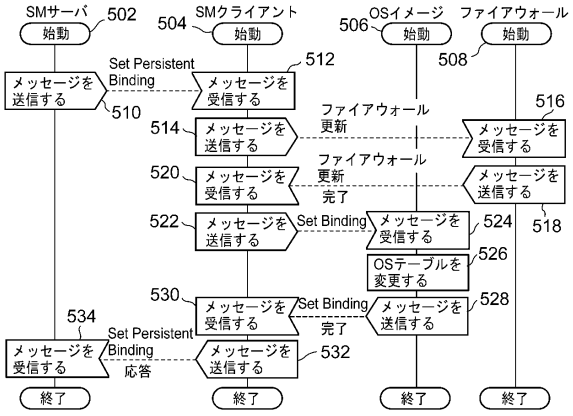
【図３】


開始
SMコア・エンジンが要求を作成する
通信モジュールが要求のターゲットを決定する
SANまたはアダプタに関する要求
OS構成データに関する要求
通信モジュールが許可データによりSMクライアントへの通信パスを確立する
通信モジュールが許可データによりRAサーバへの通信パスを確立する
許可が有効であるかどうかをSMクライアントがチェックする
許可が有効であるかどうかをRAサーバがチェックする
YES
NO
アダプタおよびSAN属性を照会し設定することにより応答を作成する
リジェクト応答を作成する
OS構成操作により応答を作成する
通信システムに応答を送信する
通信システムがSMコア・エンジンに応答を転送する
SMコア・エンジンが応答を処理する
終了

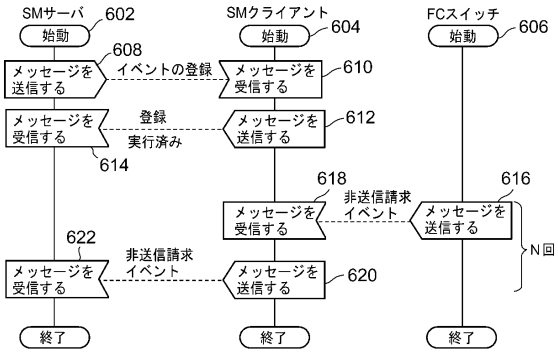
【図４】


SMサーバ
SMクライアント
FCスイッチ
始動
HBA_API: LIRRを送信する
メッセージを送信する
メッセージを受信する
ELS LIRR
登録を変更する
応答
終了

【図５】


SMサーバ
SMクライアント
OSイメージ
ファイアウォール
始動
Set Persistent Binding
メッセージを送信する
メッセージを受信する
ファイアウォール更新
Set Binding
完了
OSテーブルを変更する
応答
終了

【図６】


SMサーバ
602
始動
SMクライアント
始動
604
FCスイッチ
始動
606
608
メッセージを送信する
イベントの登録
メッセージを受信する
610
メッセージを受信する
登録
実行済み
メッセージを送信する
612
614
618
非送信請求イベント
616
メッセージを受信する
メッセージを送信する
N回
622
非送信請求イベント
メッセージを受信する
メッセージを送信する
620
終了
終了
終了

フロントページの続き

(74)代理人 100086243
弁理士 坂口 博
(72)発明者 ライシュ、クリシュトフ
ドイツ連邦共和国７０８３９ ゲルリンゲン フリードリヒ・シェファルト・シュトラッセ ２１

審査官 木村 貴俊

(56)参考文献 特開２００２－３３５２６５（ＪＰ，Ａ）
特開２００２－０６３０６３（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 3/06- 3/08
G06F 12/00-12/16
G06F 13/10-13/14