(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2019-503598
(P2019-503598A)
(43)公表日　平成31年2月7日(2019.2.7)

(51)Int.Cl.			F I			テーマコード（参考）
H04L	***12/931***	***(2013.01)***	H04L	12/931		5K030
H04L	***12/70***	***(2013.01)***	H04L	12/70	100Z	

審査請求 未請求　予備審査請求 未請求　（全 42 頁）

(21)出願番号	特願2018-504731 (P2018-504731)
(86)(22)出願日	平成29年1月26日 (2017.1.26)
(85)翻訳文提出日	平成30年1月29日 (2018.1.29)
(86)国際出願番号	PCT/US2017/015156
(87)国際公開番号	WO2017/132392
(87)国際公開日	平成29年8月3日 (2017.8.3)
(31)優先権主張番号	62/287,704
(32)優先日	平成28年1月27日 (2016.1.27)
(33)優先権主張国	米国 (US)
(31)優先権主張番号	15/412,972
(32)優先日	平成29年1月23日 (2017.1.23)
(33)優先権主張国	米国 (US)
(31)優先権主張番号	15/415,497
(32)優先日	平成29年1月25日 (2017.1.25)
(33)優先権主張国	米国 (US)

(71)出願人　502303739
オラクル・インターナショナル・コーポレイション
アメリカ合衆国カリフォルニア州９４０６５レッドウッド・シティー，オラクル・パークウェイ５００
(74)代理人　110001195
特許業務法人深見特許事務所
(72)発明者　ヨンセン，ビョルン・ダグ
ノルウェー、０６８７　オスロ、ビルベルクグレンダ、９
(72)発明者　スリニバサン，アルビンド
アメリカ合衆国、９５１２９　カリフォルニア州、サン・ノゼ、ハッピー・バレー・アベニュ、１０７５

最終頁に続く

(54)【発明の名称】高性能コンピューティング環境においてスケーラブルなビットマップに基づくＰ＿Ｋｅｙテーブルをサポートするためのシステムおよび方法

(57)【要約】
高性能コンピューティング環境においてスケーラブルなビットマップに基づくＰ＿Ｋｅｙテーブルをサポートするためのシステムおよび方法。ある方法は、１つ以上のスイッチ、複数のホストチャネルアダプタ、および複数のエンドノードを備える少なくとも１つのサブネットを提供することができる。この方法は、複数のエンドノードを複数のパーティションの少なくとも１つに関連付けることができ、複数のパーティションの各々はＰ＿Ｋｅｙ値に関連付けられる。この方法は、１つ以上のスイッチの各々を、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることができる。この方法は、ホストチャネルアダプタの各々を、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることができる。

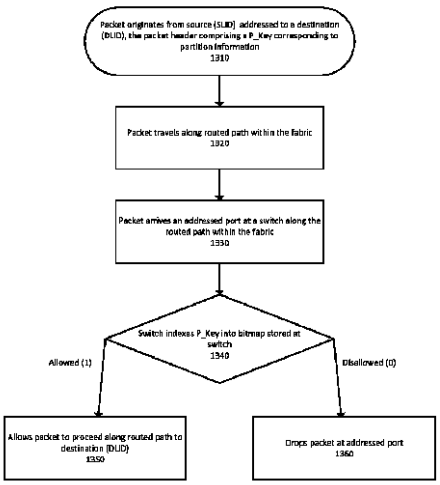

FIGURE 13

【特許請求の範囲】
【請求項１】
高性能コンピューティング環境においてスケーラブルなビットマップに基づくＰ＿Ｋｅｙテーブルをサポートするためのシステムであって、
１つ以上のマイクロプロセッサと、
少なくとも１つのサブネットとを備え、前記少なくとも１つのサブネットは、
１つ以上のスイッチを含み、前記１つ以上のスイッチは少なくともリーフスイッチを含み、前記１つ以上のスイッチの各々は、複数のスイッチポートを含み、前記少なくとも１つのサブネットはさらに、
複数のホストチャネルアダプタを含み、各ホストチャネルアダプタは、少なくとも１つのホストチャネルアダプタポートを含み、前記少なくとも１つのサブネットはさらに、
複数のエンドノードを含み、前記複数のエンドノードの各々は、前記複数のホストチャネルアダプタのうちの少なくとも１つのホストチャネルアダプタに関連付けられ、
前記複数のエンドノードの各々は、複数のパーティションのうちの少なくとも１つに関連付けられ、
前記複数のパーティションの各々は、Ｐ＿Ｋｅｙ値に関連付けられ、
前記複数のスイッチポートのうちのあるスイッチポートは、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けられ、
前記複数のホストチャネルアダプタポートのうちのあるホストチャネルアダプタポートは、前記複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けられる、システム。
【請求項２】
前記１つ以上のスイッチの１つまたは前記複数のホストチャネルアダプタの１つで実行されるサブネットマネージャをさらに備え、
前記サブネットマネージャは、前記１つ以上のスイッチの各々上の複数のポートを通る許可されたトラフィックおよび許可されないトラフィックを判断する、請求項１に記載のシステム。
【請求項３】
前記サブネットマネージャは、前記１つ以上のスイッチの各々上の複数のポートの各々を通る許可されたトラフィックおよび許可されないトラフィックの前記判断に基づいて、前記ビットマップに基づくＰ＿Ｋｅｙテーブルの各々を構成する、請求項２に記載のシステム。
【請求項４】
前記１つ以上のスイッチのうちのあるスイッチが、Ｐ＿Ｋｅｙ値を含む少なくともヘッダを含むパケットを、アドレス指定されたポートで受信すると、前記１つ以上のスイッチのうちの前記あるスイッチは、許可されたまたは許可されないインジケータを受信するよう、前記パケットの前記ヘッダに含まれる前記Ｐ＿Ｋｅｙ値を、関連付けられるビットマップに基づくＰ＿Ｋｅｙ値で索引付けする、請求項１に記載のシステム。
【請求項５】
許可された値を受信すると、前記スイッチは、前記パケットが前記アドレス指定されたポートを通過することを可能にする、請求項４に記載のシステム。
【請求項６】
許可されない値を受信すると、前記スイッチは、前記パケットを前記アドレス指定されたポートでドロップする、請求項４に記載のシステム。
【請求項７】
前記１つ以上のサブネットは、２つ以上のサブネットを含み、前記２つ以上のサブネットの各々は、前記２つ以上のサブネットの各々において少なくとも１つのルータによって相互接続される、先行する請求項のいずれか１つに記載のシステム。
【請求項８】

高性能コンピューティング環境においてスケーラブルなビットマップに基づくＰ＿Ｋｅｙテーブルをサポートするための方法であって、
１つ以上のマイクロプロセッサを含む１つ以上のコンピュータに少なくとも１つのサブネットを提供することを備え、前記少なくとも１つのサブネットは、
１つ以上のスイッチを含み、前記１つ以上のスイッチは少なくともリーフスイッチを含み、前記１つ以上のスイッチの各々は、複数のスイッチポートを含み、前記少なくとも１つのサブネットはさらに、
複数のホストチャネルアダプタを含み、各ホストチャネルアダプタは、少なくとも１つのホストチャネルアダプタポートを含み、前記少なくとも１つのサブネットはさらに、
複数のエンドノードを含み、前記複数のエンドノードの各々は、前記複数のホストチャネルアダプタのうちの少なくとも１つのホストチャネルアダプタに関連付けられ、前記方法はさらに、
前記複数の物理ホスト仮想マシンの各々を複数のパーティションの少なくとも１つに関連付けることを備え、前記複数のパーティションの各々はＰ＿Ｋｅｙ値に関連付けられ、前記方法はさらに、
前記複数のスイッチポートのうちのあるスイッチポートを、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることと、
前記複数のホストチャネルアダプタポートのうちのあるホストチャネルアダプタポートを、前記複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることとを備える、高性能コンピューティング環境においてスケーラブルなビットマップに基づくＰ＿Ｋｅｙテーブルをサポートするための方法。

【請求項９】
前記１つ以上のマイクロプロセッサを含む前記１つ以上のコンピュータにおいて、
前記１つ以上のスイッチの１つまたは前記複数のホストチャネルアダプタの１つで実行されるサブネットマネージャを提供することと、
前記サブネットマネージャによって、前記１つ以上のスイッチの各々上の複数のポートを通る許可されたトラフィックおよび許可されないトラフィックを判断することとをさらに備える、請求項８に記載の方法。

【請求項１０】
前記サブネットマネージャによって、前記１つ以上のスイッチの各々上の複数のポートの各々を通る許可されたトラフィックおよび許可されないトラフィックの前記判断に基づいて、前記ビットマップに基づくＰ＿Ｋｅｙテーブルの各々を構成することをさらに備える、請求項９に記載の方法。

【請求項１１】
前記１つ以上のスイッチのうちのあるスイッチが、Ｐ＿Ｋｅｙ値を含む少なくともヘッダを含むパケットを、アドレス指定されたポートで受信すると、許可されたまたは許可されないインジケータを受信するよう、前記１つ以上のスイッチのうちの前記あるスイッチによって、前記パケットの前記ヘッダに含まれる前記Ｐ＿Ｋｅｙ値を、関連付けられるビットマップに基づくＰ＿Ｋｅｙ値で索引付けすることをさらに備える、請求項８～１０のいずれか１つに記載の方法。

【請求項１２】
許可された値を受信すると、前記スイッチによって、前記パケットが前記アドレス指定されたポートを通過することを可能にすることをさらに備える、請求項１１に記載の方法。

【請求項１３】
許可されない値を受信すると、前記スイッチによって、前記パケットを前記アドレス指定されたポートでドロップすることをさらに備える、請求項１１に記載の方法。

【請求項１４】

前記１つ以上のサブネットは、２つ以上のサブネットを含み、前記２つ以上のサブネットの各々は、前記２つ以上のサブネットの各々において少なくとも１つのルータによって相互接続される、請求項８～１３のいずれか１つに記載の方法。

【請求項１５】

高性能コンピューティング環境においてスケーラブルなビットマップに基づくＰ＿Ｋｅｙテーブルをサポートするための命令をそこに記憶して含む、非一時的なコンピュータ可読記憶媒体であって、前記命令は、１つ以上のコンピュータによって読み取られ実行されると、前記１つ以上のコンピュータに、

１つ以上のマイクロプロセッサを含む１つ以上のコンピュータに少なくとも１つのサブネットを提供することを含むステップを実行させ、前記少なくとも１つのサブネットは、

１つ以上のスイッチを含み、前記１つ以上のスイッチは少なくともリーフスイッチを含み、前記１つ以上のスイッチの各々は、複数のスイッチポートを含み、前記少なくとも１つのサブネットはさらに、

複数のホストチャネルアダプタを含み、各ホストチャネルアダプタは、少なくとも１つのホストチャネルアダプタポートを含み、前記少なくとも１つのサブネットはさらに、

複数のエンドノードを含み、前記複数のエンドノードの各々は、前記複数のホストチャネルアダプタのうちの少なくとも１つのホストチャネルアダプタに関連付けられ、前記命令は、さらに、１つ以上のコンピュータによって読み取られ実行されると、前記１つ以上のコンピュータに、

前記複数のエンドノードの各々を複数のパーティションの少なくとも１つに関連付けることを含むステップを実行させ、前記複数のパーティションの各々はＰ＿Ｋｅｙ値に関連付けられ、前記命令は、さらに、１つ以上のコンピュータによって読み取られ実行されると、前記１つ以上のコンピュータに、

前記複数のスイッチポートのうちのあるスイッチポートを、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることと、

前記複数のホストチャネルアダプタポートのうちのあるホストチャネルアダプタポートを、前記複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることとを含むステップを実行させる、非一時的なコンピュータ可読記憶媒体。

【請求項１６】

前記ステップは、さらに、前記１つ以上のマイクロプロセッサを含む前記１つ以上のコンピュータにおいて、

前記１つ以上のスイッチの１つまたは前記複数のホストチャネルアダプタの１つで実行されるサブネットマネージャを提供することと、

前記サブネットマネージャによって、前記１つ以上のスイッチの各々上の複数のポートを通る許可されたトラフィックおよび許可されないトラフィックを判断することとを含む、請求項１５に記載の非一時的なコンピュータ可読記憶媒体。

【請求項１７】

前記ステップは、さらに、前記サブネットマネージャによって、前記１つ以上のスイッチの各々上の複数のポートの各々を通る許可されたトラフィックおよび許可されないトラフィックの前記判断に基づいて、前記ビットマップに基づくＰ＿Ｋｅｙテーブルの各々を構成することを含む、請求項１６に記載の非一時的なコンピュータ可読記憶媒体。

【請求項１８】

前記ステップは、さらに、前記１つ以上のスイッチのうちのあるスイッチが、Ｐ＿Ｋｅｙ値を含む少なくともヘッダを含むパケットを、アドレス指定されたポートで受信すると、許可されたまたは許可されないインジケータを受信するよう、前記１つ以上のスイッチのうちの前記あるスイッチによって、前記パケットの前記ヘッダに含まれる前記Ｐ＿Ｋｅｙ値を、関連付けられるビットマップに基づくＰ＿Ｋｅｙ値で索引付けすることを含む、請求項１５～１７のいずれか１つに記載の一時的なコンピュータ可読記憶媒体。

【請求項１９】
前記ステップは、さらに、許可された値を受信すると、前記スイッチによって、前記パケットが前記アドレス指定されたポートを通過することを可能にすることを含む、請求項１８に記載の非一時的なコンピュータ可読記憶媒体。
【請求項２０】
前記ステップは、さらに、許可されない値を受信すると、前記スイッチによって、前記パケットを前記アドレス指定されたポートでドロップすることを含む、請求項１８に記載の非一時的なコンピュータ可読記憶媒体。
【請求項２１】
コンピュータシステムによって実行されると、前記コンピュータシステムに請求項８～１４のいずれか１つに記載の方法を実行させる、機械可読フォーマットにおけるプログラム命令を含む、コンピュータプログラム。
【請求項２２】
非一時的な機械可読データ記憶媒体に記憶される請求項２１のコンピュータプログラムを備えるコンピュータプログラム製品。
【発明の詳細な説明】
【技術分野】
【０００１】
著作権表示：
この特許文献の開示の一部は、著作権保護の対象となる資料を含む。著作権保有者は、この特許文献または特許開示の、それが特許商標庁の特許ファイルまたは記録に現れているとおりの、何人による複写複製にも異議を唱えないが、それ以外の場合にはすべての著作権をどのようなものであろうと所有する。
【０００２】
発明の分野：
本発明は、一般にコンピュータシステムに関し、特に、高性能コンピューティング環境においてスケーラブルビットマップに基づくＰ＿Ｋｅｙテーブルをサポートすることに関する。
【背景技術】
【０００３】
背景：
導入されるクラウドコンピューティングアーキテクチャがより大規模になるのに応じて、従来のネットワークおよびストレージに関する性能および管理の障害が深刻な問題になってきている。クラウドコンピューティングファブリックのための基礎としてインフィニバンド（登録商標）（InfiniBand：ＩＢ）技術などの高性能な無損失相互接続を用いることへの関心がますます高まってきている。これは、本発明の実施形態が対応するように意図された一般領域である。
【発明の概要】
【課題を解決するための手段】
【０００４】
概要：
本明細書では、高性能コンピューティング環境においてスケーラブルなビットマップに基づくＰ＿Ｋｅｙテーブルをサポートするためのシステムおよび方法を説明する。ある例示的方法は、１つ以上のスイッチを含む少なくとも１つのサブネットを提供し得、１つ以上のスイッチは少なくともリーフスイッチを含み、１つ以上のスイッチの各々は、複数のスイッチポートを含み、少なくとも１つのサブネットはさらに、複数のホストチャネルアダプタを含み、ホストチャネルアダプタの１つ以上は、少なくとも１つの仮想機能、少なくとも１つの仮想スイッチ、および少なくとも１つの物理機能を含み、複数のホストチャネルアダプタの各々は、複数のホストチャネルアダプタポートを含み、複数のホストチャネルアダプタは、１つ以上のスイッチを介して相互接続され、少なくとも１つのサブネッ

トはさらに、複数の物理ホストおよびハイパーバイザを含み、複数の物理ホストおよびハイパーバイザの各々は、複数のホストチャネルアダプタのうちの少なくとも１つのホストチャネルアダプタに関連付けられ、少なくとも１つのサブネットはさらに、複数の仮想マシンを含み、複数の仮想マシンの各々は、少なくとも１つの仮想機能に関連付けられている。この方法は、複数の物理ホストおよび仮想マシンの各々を複数のパーティションの少なくとも１つに関連付けることができ、複数のパーティションの各々はＰ＿Ｋｅｙ値に関連付けられる。この方法は、１つ以上のスイッチポートの各々を、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることができる。この方法は、ホストチャネルアダプタポートの各々を、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることができる。

【０００５】

本明細書では、高性能コンピューティング環境においてビットマップに基づくハードウェア実現例を使用してレガシーＰ＿Ｋｅｙテーブル抽象化をサポートするためのシステムおよび方法を説明する。ある例示的方法は、１つ以上のマイクロプロセッサを含む１つ以上のコンピュータに少なくとも１つのサブネットを提供することができ、少なくとも１つのサブネットは、１つ以上のスイッチを含み、１つ以上のスイッチは少なくともリーフスイッチを含み、１つ以上のスイッチの各々は、複数のスイッチポートを含み、少なくとも１つのサブネットはさらに、複数のホストチャネルアダプタを含み、各ホストチャネルアダプタは、少なくとも１つのホストチャネルアダプタポートを含み、複数のホストチャネルアダプタは、１つ以上のスイッチを介して相互接続され、少なくとも１つのサブネットはさらに、複数のエンドノードを含み、複数のエンドノードの各々は、複数のホストチャネルアダプタのうちの少なくとも１つのホストチャネルアダプタに関連付けられる。この方法は、エンドノードの各々を複数のパーティションの少なくとも１つに関連付けることができ、複数のパーティションの各々はＰ＿Ｋｅｙ値に関連付けられる。この方法は、１つ以上のスイッチポートの各々を、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることができる。この方法は、ホストチャネルアダプタポートの各々を、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることができる。この方法は、複数のビットマップに基づくＰ＿Ｋｅｙテーブルの各々を、仮想Ｐ＿Ｋｅｙテーブルに関連付けることができる。

【０００６】

一実施形態によれば、複数のホストチャネルアダプタの１つ以上は、少なくとも１つの仮想機能、少なくとも１つの仮想スイッチ、および少なくとも１つの物理機能を含むことができる。複数のエンドノードは、物理ホスト、仮想マシン、または物理ホストと仮想マシンとの組み合わせを含むことができ、仮想マシンは、少なくとも１つの仮想機能に関連付けられる。

【図面の簡単な説明】

【０００７】

【図１】一実施形態によるインフィニバンド環境の一例を示す図である。

【図２】一実施形態による、パーティショニングされたクラスタ環境を示す図である。

【図３】一実施形態による、ネットワーク環境におけるツリートポロジを示す図である。

【図４】一実施形態に従った例示的な共有ポートアーキテクチャを示す図である。

【図５】一実施形態に従った例示的なｖＳｗｉｔｃｈアーキテクチャを示す図である。

【図６】一実施形態に従った例示的なｖＰｏｒｔアーキテクチャを示す図である。

【図７】一実施形態に従った、ＬＩＤが予めポピュレートされた例示的なｖＳｗｉｔｃｈアーキテクチャを示す図である。

【図８】一実施形態に従った、動的ＬＩＤ割当てがなされた例示的なｖＳｗｉｔｃｈアーキテクチャを示す図である。

【図９】一実施形態に従った、動的ＬＩＤ割当てがなされかつＬＩＤが予めポピュレート

されているｖＳｗｉｔｃｈを備えた例示的なｖＳｗｉｔｃｈアーキテクチャを示す図である。
【図１０】一実施形態による例示的なマルチサブネットインフィニバンドファブリックを示す。
【図１１】一実施形態による、パーティション分離のための連想テーブルを有する例示的なインフィニバンドファブリックを示す。
【図１２】一実施形態による、パーティション分離のためのビットマップを有する例示的なインフィニバンドファブリックを示す。
【図１３】一実施形態による、高性能コンピューティング環境においてスケーラブルなビットマップに基づくＰ＿Ｋｅｙテーブルをサポートするための方法のフローチャートである。
【図１４】一実施形態による、パーティション分離のためのビットマップと、レガシー管理エンティティのためのＰ＿Ｋｅｙ抽象化とを有する例示的なファブリックを示す。
【図１５】一実施形態による、パーティション分離のためのビットマップと、レガシー管理エンティティのためのＰ＿Ｋｅｙ抽象化とを有する例示的なファブリックを示す。
【図１６】一実施形態による、高性能コンピューティング環境においてスケーラブルなビットマップに基づくＰ＿Ｋｅｙテーブルをサポートするための方法のフローチャートである。
【図１７】一実施形態による、高性能コンピューティング環境においてビットマップに基づくハードウェア実現例を使用してレガシーＰ＿Ｋｅｙテーブル抽象化をサポートするための方法のフローチャートである。
【発明を実施するための形態】
【０００８】
詳細な説明：
本発明は、同様の参照番号が同様の要素を指している添付図面の図において、限定のためではなく例示のために説明されている。なお、この開示における「ある」または「１つの」または「いくつかの」実施形態への参照は必ずしも同じ実施形態に対するものではなく、そのような参照は少なくとも１つを意味する。特定の実現例が説明されるが、これらの特定の実現例が例示的な目的のためにのみ提供されることが理解される。当業者であれば、他の構成要素および構成が、この発明の範囲および精神から逸脱することなく使用され得ることを認識するであろう。
【０００９】
図面および詳細な説明全体にわたって同様の要素を示すために、共通の参照番号が使用され得る。したがって、ある図で使用される参照番号は、要素が別のところで説明される場合、そのような図に特有の詳細な説明において参照される場合もあり、または参照されない場合もある。
【００１０】
本明細書では、高性能コンピューティング環境においてビットマップに基づくハードウェア実現例を使用して構成可能なレガシーＰ＿Ｋｅｙテーブル抽象化をサポートするシステムおよび方法を説明する。
【００１１】
本発明の以下の説明は、高性能ネットワークのための例としてインフィニバンド^{TM}（ＩＢ）ネットワークを使用する。以下の説明を通して、インフィニバンド^{TM}規格（様々に、インフィニバンド規格、ＩＢ規格、またはレガシーＩＢ規格とも呼ばれる）を参照することができる。このような参照は、引用によりその全体が本明細書に援用される、http://www.inifinibandta.orgで入手可能なインフィニバンド（登録商標）トレード・アソシエーション・アーキテクチャ規格、第１巻、バージョン１．３（２０１５年３月リリース）を参照すると理解される。他のタイプの高性能ネットワークが何ら限定されることなく使用され得ることが、当業者には明らかであるだろう。以下の説明ではまた、ファブリックトポロジーについての一例として、ファットツリートポロジーを使用する。他のタイプ

のファブリックトポロジーが何ら限定されることなく使用され得ることが当業者には明らかであるだろう。

【００１２】

現代（たとえばExascale（エクサスケール）時代）におけるクラウドの要求を満たすために、仮想マシンがリモート・ダイレクト・メモリ・アクセス（Remote Direct Memory Access：ＲＤＭＡ）などの低オーバーヘッドネットワーク通信パラダイムを利用できることが望ましい。ＲＤＭＡはＯＳスタックをバイパスし、ハードウェアと直接通信することで、シングルルートＩ／Ｏ仮想化（Single-Root I/O Virtualization：ＳＲ－ＩＯＶ）ネットワークアダプタのようなパススルー技術が使用可能となる。一実施形態に従うと、高性能な無損失相互接続ネットワークにおける適用可能性のために、仮想スイッチ（virtual switch：ｖＳｗｉｔｃｈ）ＳＲ－ＩＯＶアーキテクチャを提供することができる。ライブマイグレーションを実際に選択できるようにするためにネットワーク再構成時間が重要となるので、ネットワークアーキテクチャに加えて、スケーラブルであるとともにトポロジーに依存しない動的な再構成メカニズムを提供することができる。

【００１３】

一実施形態に従うと、さらには、ｖＳｗｉｔｃｈを用いる仮想化された環境のためのルーティング戦略を提供することができ、ネットワークトポロジー（たとえばファットツリートポロジー）のための効率的なルーティングアルゴリズムを提供することができる。動的な再構成メカニズムは、ファットツリーにおいて課されるオーバーヘッドを最小限にするためにさらに調整することができる。

【００１４】

本発明の一実施形態に従うと、仮想化は、クラウドコンピューティングにおける効率的なリソース利用および融通性のあるリソース割当てに有益であり得る。ライブマイグレーションは、アプリケーションにトランスペアレントな態様で物理サーバ間で仮想マシン（virtual machine：ＶＭ）を移動させることによってリソース使用を最適化することを可能にする。このため、仮想化は、ライブマイグレーションによる統合、リソースのオン・デマンド・プロビジョニングおよび融通性を可能にし得る。

【００１５】

インフィニバンド（登録商標）

インフィニバンド（ＩＢ）は、インフィニバンド・トレード・アソシエーション（InfiniBand™Trade Association）によって開発されたオープン標準無損失ネットワーク技術である。この技術は、特に高性能コンピューティング（high-performance computing：ＨＰＣ）アプリケーションおよびデータセンタを対象とする、高スループットおよび少ない待ち時間の通信を提供するシリアルポイントツーポイント全二重相互接続（serial point-to-point full-duplex interconnect）に基づいている。

【００１６】

インフィニバンド・アーキテクチャ（InfiniBand Architecture：ＩＢＡ）は、２層トポロジー分割をサポートする。低層では、ＩＢネットワークはサブネットと呼ばれ、１つのサブネットは、スイッチおよびポイントツーポイントリンクを使用して相互接続される一組のホストを含み得る。より高いレベルでは、１つのＩＢファブリックは、ルータを使用して相互接続され得る１つ以上のサブネットを構成する。

【００１７】

１つのサブネット内で、ホストは、スイッチおよびポイントツーポイントリンクを使用して接続され得る。加えて、サブネットにおける指定されたデバイス上に存在する、１つのマスター管理エンティティ、すなわちサブネットマネージャ（subnet manager：ＳＭ）があり得る。サブネットマネージャは、ＩＢサブネットを構成し、起動し、維持する役割を果たす。加えて、サブネットマネージャ（ＳＭ）は、ＩＢファブリックにおいてルーティングテーブル計算を行なう役割を果たし得る。ここで、たとえば、ＩＢネットワークのルーティングは、ローカルサブネットにおけるすべての送信元と宛先とのペア間の適正な負荷バランシングを目標とする。

【００１８】
サブネット管理インターフェイスを通して、サブネットマネージャは、サブネット管理パケット（subnet management packet：ＳＭＰ）と呼ばれる制御パケットを、サブネット管理エージェント（subnet management agent：ＳＭＡ）と交換する。サブネット管理エージェントは、すべてのＩＢサブネットデバイス上に存在する。ＳＭＰを使用することにより、サブネットマネージャは、ファブリックを発見し、エンドノードおよびスイッチを構成し、ＳＭＡから通知を受信することができる。
【００１９】
一実施形態によれば、ＩＢネットワークにおけるサブネット内のルーティングは、スイッチに格納されたＬＦＴに基づき得る。ＬＦＴは、使用中のルーティングメカニズムに従って、ＳＭによって計算される。サブネットでは、エンドノード上のホストチャネルアダプタ（Host Channel Adapter：ＨＣＡ）ポートおよびスイッチが、ローカル識別子（ＬＩＤ）を使用してアドレス指定される。ＬＦＴにおける各エントリは、宛先ＬＩＤ（destination LID：ＤＬＩＤ）と出力ポートとからなる。テーブルにおけるＬＩＤごとに１つのエントリのみがサポートされる。パケットがあるスイッチに到着すると、その出力ポートは、そのスイッチのフォワーディングテーブルにおいてＤＬＩＤを検索することによって判断される。所与の送信元－宛先ペア（ＬＩＤペア）間のネットワークにおいてパケットは同じ経路を通るため、ルーティングは決定論的である。
【００２０】
一般に、マスターサブネットマネージャを除く他のすべてのサブネットマネージャは、耐故障性のために待機モードで作動する。しかしながら、マスターサブネットマネージャが故障した状況では、待機中のサブネットマネージャによって、新しいマスターサブネットマネージャが取り決められる。マスターサブネットマネージャはまた、サブネットの周期的なスイープ（sweep）を行なってあらゆるトポロジー変化を検出し、それに応じてネットワークを再構成する。
【００２１】
さらに、サブネット内のホストおよびスイッチは、ローカル識別子（ＬＩＤ）を用いてアドレス指定され得るとともに、単一のサブネットは４９１５１個のユニキャストＬＩＤに制限され得る。サブネット内で有効なローカルアドレスであるＬＩＤの他に、各ＩＢデバイスは、６４ビットのグローバル一意識別子（global unique identifier：ＧＵＩＤ）を有し得る。ＧＵＩＤは、ＩＢレイヤー３（Ｌ３）アドレスであるグローバル識別子（global identifier：ＧＩＤ）を形成するために使用され得る。
【００２２】
ＳＭは、ネットワーク初期化時間に、ルーティングテーブル（すなわち、サブネット内のノードの各ペア間の接続／ルート）を計算し得る。さらに、トポロジーが変化するたびに、ルーティングテーブルは、接続性および最適性能を確実にするために更新され得る。通常動作中、ＳＭは、トポロジー変化をチェックするためにネットワークの周期的なライトスイープ（light sweep）を実行し得る。ライトスイープ中に変化が発見された場合、または、ネットワーク変化を信号で伝えるメッセージ（トラップ）をＳＭが受信した場合、ＳＭは、発見された変化に従ってネットワークを再構成し得る。
【００２３】
たとえば、ＳＭは、リンクがダウンした場合、デバイスが追加された場合、またはリンクが除去された場合など、ネットワークトポロジーが変化する場合に、ネットワークを再構成し得る。再構成ステップは、ネットワーク初期化中に行なわれるステップを含み得る。さらに、再構成は、ネットワーク変化が生じたサブネットに制限されるローカルスコープを有し得る。また、ルータを用いる大規模ファブリックのセグメント化は、再構成スコープを制限し得る。
【００２４】
一実施形態に従ったインフィニバンド環境１００の例を示す図１に、インフィニバンドファブリックの一例を示す。図１に示す例では、ノードＡ１０１～Ｅ１０５は、インフィ

ニバンドファブリック１２０を使用して、それぞれのホストチャネルアダプタ１１１～１１５を介して通信する。一実施形態に従うと、さまざまなノード（たとえばノードＡ１０１～Ｅ１０５）はさまざまな物理デバイスによって表わすことができる。一実施形態に従うと、さまざまなノード（たとえばノードＡ１０１～Ｅ１０５）は仮想マシンなどのさまざまな仮想デバイスによって表わすことができる。

【００２５】

インフィニバンドにおけるパーティショニング

一実施形態によれば、ＩＢネットワークは、ネットワークファブリックを共有するシステムの論理グループの分離をもたらすためにセキュリティメカニズムとしてパーティショニングをサポートし得る。ファブリックにおけるノード上の各ＨＣＡポートは、１つ以上のパーティションのメンバであり得る。パーティションメンバーシップは、ＳＭの一部であり得る集中型パーティションマネージャによって管理される。ＳＭは、各ポートに関するパーティションメンバーシップ情報を、１６ビットのパーティションキー（partition key：Ｐ＿キー）のテーブルとして構成することができる。ＳＭはまた、これらのポートを介してデータトラフィックを送信または受信するエンドノードに関連付けられたＰ＿Ｋｅｙ情報を含むパーティション実施テーブルを用いて、スイッチポートおよびルータポートを構成することができる。加えて、一般的な場合には、スイッチポートのパーティションメンバーシップは、（リンクに向かう）出口方向に向かってポートを介してルーティングされたＬＩＤに間接的に関連付けられたすべてのメンバーシップの集合を表わし得る。

【００２６】

一実施形態によれば、パーティションは、あるグループのメンバが同じ論理グループの他のメンバとしか通信できないような、ポートの論理グループである。ホストチャネルアダプタ（ＨＣＡ）およびスイッチでは、パーティションメンバシップ情報を使用してパケットをフィルタリングして分離を実行できる。無効なパーティショニング情報を持つパケットは、そのパケットが着信ポートに到着するとすぐにドロップすることができる。パーティショニングされたＩＢシステムでは、パーティションを使用してテナントクラスタを作成できる。パーティションの適所における実施で、ノードは異なるテナントクラスタに属する他のノードと通信することはできない。このようにして、侵害されたテナントノードまたは悪意のあるテナントノードが存在する場合でも、システムのセキュリティを保証することができる。

【００２７】

一実施形態によれば、ノード間の通信のために、管理キューペア（ＱＰ０およびＱＰ１）を除き、キューペア（Queue Pair：ＱＰ）およびエンドツーエンドコンテキスト（End-to-End context：ＥＥＣ）を特定のパーティションに割当てることができる。次に、Ｐ＿キー情報を、送信されたすべてのＩＢトランスポートパケットに追加することができる。パケットがＨＣＡポートまたはスイッチに到着すると、そのＰ＿キー値を、ＳＭによって構成されたテーブルに対して確認することができる。無効のＰ＿キー値が見つかった場合、そのパケットは直ちに廃棄される。このように、通信は、パーティションを共有するポート間でのみ許可される。

【００２８】

ＩＢパーティションのある例が、図２に示されており、それは、一実施形態による、パーティショニングされたクラスタ環境を示している。図２に示す例では、ノードＡ～Ｅ１０１～１０５は、インフィニバンドファブリック１２０を使用して、それぞれのホストチャネルアダプタ１１１～１１５を介して通信する。ノードＡ～Ｅは、パーティション、すなわちパーティション１，１３０、パーティション２，１４０、およびパーティション３，１５０に配置される。パーティション１は、ノードＡ１０１およびノードＤ１０４を含む。パーティション２は、ノードＡ１０１、ノードＢ１０２、およびノードＣ１０３を含む。パーティション３は、ノードＣ１０３およびノードＥ１０５を含む。パーティションの配置のため、ノードＤ１０４とノードＥ１０５とは、これらのノードがパーティションを共有しないので、通信することができない。一方、例えば、ノードＡ１０１とノードＣ

１０３とは、これらのノードが両方ともパーティション２，１４０のメンバであるため、通信が許可される。

【００２９】

インフィニバンドにおける仮想マシン

過去１０年の間に、ハードウェア仮想化サポートによってＣＰＵオーバーヘッドが実質的に排除され、メモリ管理ユニットを仮想化することによってメモリオーバーヘッドが著しく削減され、高速ＳＡＮストレージまたは分散型ネットワークファイルシステムの利用によってストレージオーバーヘッドが削減され、シングルルートＩ／Ｏ仮想化（Single Root Input/Output Virtualization：ＳＲ－ＩＯＶ）のようなデバイス・パススルー技術を使用することによってネットワークＩ／Ｏオーバーヘッドが削減されてきたことに応じて、仮想化された高性能コンピューティング（High Performance Computing：ＨＰＣ）環境の将来見通しが大幅に改善されてきた。現在では、クラウドが、高性能相互接続ソリューションを用いて仮想ＨＰＣ（virtual HPC：ｖＨＰＣ）クラスタに対応し、必要な性能を提供することができる。

【００３０】

しかしながら、インフィニバンド（ＩＢ）などの無損失ネットワークと連結されたとき、仮想マシン（ＶＭ）のライブマイグレーションなどのいくつかのクラウド機能は、これらのソリューションにおいて用いられる複雑なアドレス指定およびルーティングスキームのせいで、依然として問題となる。ＩＢは、高帯域および低レイテンシを提供する相互接続ネットワーク技術であり、このため、ＨＰＣおよび他の通信集約型の作業負荷に非常によく適している。

【００３１】

ＩＢデバイスをＶＭに接続するための従来のアプローチは直接割当てされたＳＲ－ＩＯＶを利用することによるものである。しかしながら、ＳＲ－ＩＯＶを用いてＩＢホストチャネルアダプタ（ＨＣＡ）に割当てられたＶＭのライブマイグレーションを実現することは難易度の高いものであることが判明した。各々のＩＢが接続されているノードは、３つの異なるアドレス（すなわちＬＩＤ、ＧＵＩＤおよびＧＩＤ）を有する。ライブマイグレーションが発生すると、これらのアドレスのうち１つ以上が変化する。マイグレーション中のＶＭ（VM-in-migration）と通信する他のノードは接続性を失う可能性がある。これが発生すると、ＩＢサブネットマネージャ（Subnet Manager：ＳＭ）にサブネット管理（Subnet Administration：ＳＡ）経路記録クエリを送信することによって、再接続すべき仮想マシンの新しいアドレスを突きとめることにより、失われた接続を回復させるように試みることができる。

【００３２】

ＩＢは３つの異なるタイプのアドレスを用いる。第１のタイプのアドレスは１６ビットのローカル識別子（ＬＩＤ）である。少なくとも１つの固有のＬＩＤは、ＳＭによって各々のＨＣＡポートおよび各々のスイッチに割当てられる。ＬＩＤはサブネット内のトラフィックをルーティングするために用いられる。ＬＩＤが１６ビット長であるので、６５３６個の固有のアドレス組合せを構成することができ、そのうち４９１５１個（０×０００１－０×ＢＦＦＦ）だけをユニキャストアドレスとして用いることができる。結果として、入手可能なユニキャストアドレスの数は、ＩＢサブネットの最大サイズを定義することとなる。第２のタイプのアドレスは、製造業者によって各々のデバイス（たとえば、ＨＣＡおよびスイッチ）ならびに各々のＨＣＡポートに割当てられた６４ビットのグローバル一意識別子（ＧＵＩＤ）である。ＳＭは、ＨＣＡポートに追加のサブネット固有ＧＵＩＤを割当ててもよく、これは、ＳＲ－ＩＯＶが用いられる場合に有用となる。第３のタイプのアドレスは１２８ビットのグローバル識別子（ＧＩＤ）である。ＧＩＤは有効なＩＰｖ６ユニキャストアドレスであり、少なくとも１つが各々のＨＣＡポートに割当てられている。ＧＩＤは、ファブリックアドミニストレータによって割当てられたグローバルに固有の６４ビットプレフィックスと各々のＨＣＡポートのＧＵＩＤアドレスとを組合わせることによって形成される。

【００３３】

ファットツリー（Fat Tree：ＦＴｒｅｅ）トポロジーおよびルーティング

一実施形態によれば、ＩＢベースのＨＰＣシステムのいくつかは、ファットツリートポロジーを採用して、ファットツリーが提供する有用な特性を利用する。これらの特性は、各送信元宛先ペア間の複数経路の利用可能性に起因する、フルバイセクション帯域幅および固有の耐故障性を含む。ファットツリーの背後にある初期の概念は、ツリーがトポロジーのルート（root）に近づくにつれて、より利用可能な帯域幅を用いて、ノード間のより太いリンクを採用することであった。より太いリンクは、上位レベルのスイッチにおける輻輳を回避するのに役立てることができ、バイセクション帯域幅が維持される。

【００３４】

図３は、一実施形態に従った、ネットワーク環境におけるツリートポロジーの例を示す。図３に示すように、ネットワークファブリック２００において、１つ以上のエンドノード２０１～２０４が接続され得る。ネットワークファブリック２００は、複数のリーフスイッチ２１１～２１４と複数のスパインスイッチまたはルート（root）スイッチ２３１～２３４とを含むファットツリートポロジーに基づき得る。加えて、ネットワークファブリック２００は、スイッチ２２１～２２４などの１つ以上の中間スイッチを含み得る。

【００３５】

また、図３に示すように、エンドノード２０１～２０４の各々は、マルチホームノード、すなわち、複数のポートを介してネットワークファブリック２００のうち２つ以上の部分に接続される単一のノードであり得る。たとえば、ノード２０１はポートＨ１およびＨ２を含み、ノード２０２はポートＨ３およびＨ４を含み、ノード２０３はポートＨ５およびＨ６を含み、ノード２０４はポートＨ７およびＨ８を含み得る。

【００３６】

加えて、各スイッチは複数のスイッチポートを有し得る。たとえば、ルートスイッチ２３１はスイッチポート１～２を有し、ルートスイッチ２３２はスイッチポート３～４を有し、ルートスイッチ２３３はスイッチポート５～６を有し、ルートスイッチ２３４はスイッチポート７～８を有し得る。

【００３７】

一実施形態によれば、ファットツリールーティングメカニズムは、ＩＢベースのファットツリートポロジーに関して最も人気のあるルーティングアルゴリズムのうちの１つである。ファットツリールーティングメカニズムはまた、ＯＦＥＤ（Open Fabric Enterprise Distribution：ＩＢベースのアプリケーションを構築しデプロイするための標準ソフトウェアスタック）サブネットマネージャ、すなわちＯｐｅｎＳＭにおいて実現される。

【００３８】

ファットツリールーティングメカニズムの目的は、ネットワークファブリックにおけるリンクにわたって最短経路ルートを均一に広げるＬＦＴを生成することである。このメカニズムは、索引付け順序でファブリックを横断し、エンドノードの目標ＬＩＤ、ひいては対応するルートを各スイッチポートに割当てる。同じリーフスイッチに接続されたエンドノードについては、索引付け順序は、エンドノードが接続されるスイッチポートに依存し得る（すなわち、ポートナンバリングシーケンス）。各ポートについては、メカニズムはポート使用カウンタを維持することができ、新しいルートが追加されるたびに、ポート使用カウンタを使用して使用頻度が最小のポートを選択することができる。

【００３９】

一実施形態に従うと、パーティショニングされたサブネットでは、共通のパーティションのメンバではないノードは通信することを許可されない。実際には、これは、ファットツリールーティングアルゴリズムによって割当てられたルートのうちのいくつかがユーザトラフィックのために使用されないことを意味する。ファットツリールーティングメカニズムが、それらのルートについてのＬＦＴを、他の機能的経路と同じやり方で生成する場合、問題が生じる。この動作は、リンク上でバランシングを劣化させるおそれがある。なぜなら、ノードが索引付けの順序でルーティングされているからである。パーティション

に気づかずにルーティングが行なわれるため、ファットツリーでルーティングされたサブネットにより、概して、パーティション間の分離が不良なものとなる。

【００４０】

一実施形態に従うと、ファットツリーは、利用可能なネットワークリソースでスケーリングすることができる階層ネットワークトポロジーである。さらに、ファットツリーは、さまざまなレベルの階層に配置された商品スイッチを用いて容易に構築される。さらに、ｋ－ａｒｙ－ｎ－ｔｒｅｅ、拡張された一般化ファットツリー（Extended Generalized Fat-Tree：ＸＧＦＴ）、パラレルポート一般化ファットツリー（Parallel Ports Generalized Fat-Tree：ＰＧＦＴ）およびリアルライフファットツリー（Real Life Fat-Tree：ＲＬＦＴ）を含むファットツリーのさまざまな変形例が、一般に利用可能である。

【００４１】

また、ｋ－ａｒｙ－ｎ－ｔｒｅｅは、ｎレベルのファットツリーであって、k^nエンドノードと、$n \cdot k^{n-1}$スイッチとを備え、各々が２ｋポートを備えている。各々のスイッチは、ツリーにおいて上下方向に同数の接続を有している。ＸＧＦＴファットツリーは、スイッチのための異なる数の上下方向の接続と、ツリーにおける各レベルでの異なる数の接続とをともに可能にすることによって、ｋ－ａｒｙ－ｎ－ｔｒｅｅを拡張させる。ＰＧＦＴ定義はさらに、ＸＧＦＴトポロジーを拡張して、スイッチ間の複数の接続を可能にする。多種多様なトポロジーはＸＧＦＴおよびＰＧＦＴを用いて定義することができる。しかしながら、実用化するために、現代のＨＰＣクラスタにおいて一般に見出されるファットツリーを定義するために、ＰＧＦＴの制限バージョンであるＲＬＦＴが導入されている。ＲＬＦＴは、ファットツリーにおけるすべてのレベルに同じポートカウントスイッチを用いている。

【００４２】

入出力（Input/Output：Ｉ／Ｏ）仮想化

一実施形態に従うと、Ｉ／Ｏ仮想化（I/O Virtualization：ＩＯＶ）は、基礎をなす物理リソースに仮想マシン（ＶＭ）がアクセスすることを可能にすることによって、Ｉ／Ｏを利用可能にすることができる。ストレージトラフィックとサーバ間通信とを組合せると、シングルサーバのＩ／Ｏリソースにとって抗し難い高い負荷が課され、結果として、データの待機中に、バックログが発生し、プロセッサがアイドル状態になる可能性がある。Ｉ／Ｏ要求の数が増えるにつれて、ＩＯＶにより利用可能性をもたらすことができ、最新のＣＰＵ仮想化において見られる性能レベルに匹敵するように、（仮想化された）Ｉ／Ｏリソースの性能、スケーラビリティおよび融通性を向上させることができる。

【００４３】

一実施形態に従うと、Ｉ／Ｏリソースの共有を可能にして、ＶＭからリソースへのアクセスが保護されることを可能にし得るようなＩＯＶが所望される。ＩＯＶは、ＶＭにエクスポーズされる論理装置を、その物理的な実装から分離する。現在、エミュレーション、準仮想化、直接的な割当て（direct assignment：ＤＡ）、およびシングルルートＩ／Ｏ仮想化（ＳＲ－ＩＯＶ）などのさまざまなタイプのＩＯＶ技術が存在し得る。

【００４４】

一実施形態に従うと、あるタイプのＩＯＶ技術としてソフトウェアエミュレーションがある。ソフトウェアエミュレーションは分離されたフロントエンド／バックエンド・ソフトウェアアーキテクチャを可能にし得る。フロントエンドはＶＭに配置されたデバイスドライバであり得、Ｉ／Ｏアクセスをもたらすためにハイパーバイザによって実現されるバックエンドと通信し得る。物理デバイス共有比率は高く、ＶＭのライブマイグレーションはネットワークダウンタイムのわずか数ミリ秒で実現可能である。しかしながら、ソフトウェアエミュレーションはさらなる不所望な計算上のオーバーヘッドをもたらしてしまう。

【００４５】

一実施形態に従うと、別のタイプのＩＯＶ技術として直接的なデバイスの割当てがある。直接的なデバイスの割当てでは、Ｉ／ＯデバイスをＶＭに連結する必要があるが、デバ

イスはＶＭ間では共有されない。直接的な割当てまたはデバイス・パススルーは、最小限のオーバーヘッドでほぼ固有の性能を提供する。物理デバイスはハイパーバイザをバイパスし、直接、ＶＭに取付けられている。しかしながら、このような直接的なデバイスの割当ての欠点は、仮想マシン間で共有がなされないため、１枚の物理ネットワークカードが１つのＶＭと連結されるといったように、スケーラビリティが制限されてしまうことである。

【００４６】

一実施形態に従うと、シングルルートＩＯＶ（Single Root IOV：ＳＲ－ＩＯＶ）は、ハードウェア仮想化によって、物理装置がその同じ装置の複数の独立した軽量のインスタンスとして現われることを可能にし得る。これらのインスタンスは、パススルー装置としてＶＭに割当てることができ、仮想機能（Virtual Function：ＶＦ）としてアクセスすることができる。ハイパーバイザは、（１つのデバイスごとに）固有の、十分な機能を有する物理機能（Physical Function：ＰＦ）によってデバイスにアクセスする。ＳＲ－ＩＯＶは、純粋に直接的に割当てする際のスケーラビリティの問題を軽減する。しかしながら、ＳＲ－ＩＯＶによって提示される問題は、それがＶＭマイグレーションを損なう可能性があることである。これらのＩＯＶ技術の中でも、ＳＲ－ＩＯＶは、ほぼ固有の性能を維持しながらも、複数のＶＭから単一の物理デバイスに直接アクセスすることを可能にする手段を用いてＰＣＩ　Ｅｘｐｒｅｓｓ（ＰＣＩｅ）規格を拡張することができる。これにより、ＳＲ－ＩＯＶは優れた性能およびスケーラビリティを提供することができる。

【００４７】

ＳＲ－ＩＯＶは、ＰＣＩｅデバイスが、各々のゲストに１つの仮想デバイスを割当てることによって複数のゲスト間で共有することができる複数の仮想デバイスをエクスポーズすることを可能にする。各々のＳＲ－ＩＯＶデバイスは、少なくとも１つの物理機能（ＰＦ）と、１つ以上の関連付けられた仮想機能（ＶＦ）とを有する。ＰＦは、仮想マシンモニタ（virtual machine monitor：ＶＭＭ）またはハイパーバイザによって制御される通常のＰＣＩｅ機能であるのに対して、ＶＦは軽量のＰＣＩｅ機能である。各々のＶＦはそれ自体のベースアドレス（base address：ＢＡＲ）を有しており、固有のリクエスタＩＤが割当てられている。固有のリクエスタＩＤは、Ｉ／ＯメモリＭ管理ユニット（I/O memory management unit：ＩＯＭＭＵ）がさまざまなＶＦへの／からのトラフィックストリームを区別することを可能にする。ＩＯＭＭＵはまた、メモリを適用して、ＰＦとＶＦとの間の変換を中断する。

【００４８】

しかし、残念ながら、直接的デバイス割当て技術は、仮想マシンのトランスペアレントなライブマイグレーションがデータセンタ最適化のために所望されるような状況においては、クラウドプロバイダにとって障壁となる。ライブマイグレーションの本質は、ＶＭのメモリ内容がリモートハイパーバイザにコピーされるという点である。さらに、ＶＭがソースハイパーバイザにおいて中断され、ＶＭの動作が宛先において再開される。ソフトウェアエミュレーション方法を用いる場合、ネットワークインターフェイスは、それらの内部状態がメモリに記憶され、さらにコピーされるように仮想的である。このため、ダウンタイムは数ミリ秒にまで減らされ得る。

【００４９】

しかしながら、ＳＲ－ＩＯＶなどの直接的デバイス割当て技術が用いられる場合、マイグレーションはより困難になる。このような状況においては、ネットワークインターフェイスの内部状態全体は、それがハードウェアに結び付けられているのでコピーすることができない。代わりに、ＶＭに割当てられたＳＲ－ＩＯＶ　ＶＦが分離され、ライブマイグレーションが実行されることとなり、新しいＶＦが宛先において付与されることとなる。インフィニバンドおよびＳＲ－ＩＯＶの場合、このプロセスがダウンタイムを数秒のオーダでもたらす可能性がある。さらに、ＳＲ－ＩＯＶ共有型ポートモデルにおいては、ＶＭのアドレスがマイグレーション後に変化することとなり、これにより、ＳＭにオーバーヘッドが追加され、基礎をなすネットワークファブリックの性能に対して悪影響が及ぼされ

ることとなる。

【００５０】

インフィニバンドＳＲ－ＩＯＶアーキテクチャー共有ポート

さまざまなタイプのＳＲ－ＩＯＶモデル（たとえば共有ポートモデル、仮想スイッチモデルおよび仮想ポートモデル）があり得る。

【００５１】

図４は、一実施形態に従った例示的な共有ポートアーキテクチャを示す。図に示されるように、ホスト３００（たとえばホストチャネルアダプタ）はハイパーバイザ３１０と対話し得る。ハイパーバイザ３１０は、さまざまな仮想機能３３０、３４０および３５０をいくつかの仮想マシンに割当て得る。同様に、物理機能はハイパーバイザ３１０によって処理することができる。

【００５２】

一実施形態に従うと、図４に示されるような共有ポートアーキテクチャを用いる場合、ホスト（たとえばＨＣＡ）は、物理機能３２０と仮想機能３３０、３５０、３５０との間において単一の共有ＬＩＤおよび共有キュー対（Queue Pair：ＱＰ）のスペースがあるネットワークにおいて単一のポートとして現われる。しかしながら、各々の機能（すなわち、物理機能および仮想機能）はそれら自体のＧＩＤを有し得る。

【００５３】

図４に示されるように、一実施形態に従うと、さまざまなＧＩＤを仮想機能および物理機能に割当てることができ、特別のキュー対であるＱＰ０およびＱＰ１（すなわちインフィニバンド管理パケットのために用いられる専用のキュー対）が物理機能によって所有される。これらのＱＰはＶＦにも同様にエクスポーズされるが、ＶＦはＱＰ０を使用することが許可されておらず（ＶＦからＱＰ０に向かって入来するすべてのＳＭＰが廃棄され）、ＱＰ１は、ＰＦが所有する実際のＱＰ１のプロキシとして機能し得る。

【００５４】

一実施形態に従うと、共有ポートアーキテクチャは、（仮想機能に割当てられることによってネットワークに付随する）ＶＭの数によって制限されることのない高度にスケーラブルなデータセンタを可能にし得る。なぜなら、ネットワークにおける物理的なマシンおよびスイッチによってＬＩＤスペースが消費されるだけであるからである。

【００５５】

しかしながら、共有ポートアーキテクチャの欠点は、トランスペアレントなライブマイグレーションを提供することができない点であり、これにより、フレキシブルなＶＭ配置についての可能性が妨害されてしまう。各々のＬＩＤが特定のハイパーバイザに関連付けられており、かつハイパーバイザ上に常駐するすべてのＶＭ間で共有されているので、マイグレートしているＶＭ（すなわち、宛先ハイパーバイザにマイグレートする仮想マシン）は、そのＬＩＤを宛先ハイパーバイザのＬＩＤに変更させなければならない。さらに、ＱＰ０アクセスが制限された結果、サブネットマネージャはＶＭの内部で実行させることができなくなる。

【００５６】

インフィニバンドＳＲ－ＩＯＶアーキテクチャモデル－仮想スイッチ（ｖＳｗｉｔｃｈ）

図５は、一実施形態に従った例示的なｖＳｗｉｔｃｈアーキテクチャを示す。図に示されるように、ホスト４００（たとえばホストチャネルアダプタ）はハイパーバイザ４１０と対話することができ、当該ハイパーバイザ４１０は、さまざまな仮想機能４３０、４４０および４５０をいくつかの仮想マシンに割当てることができる。同様に、物理機能はハイパーバイザ４１０によって処理することができる。仮想スイッチ４１５もハイパーバイザ４０１によって処理することができる。

【００５７】

一実施形態に従うと、ｖＳｗｉｔｃｈアーキテクチャにおいては、各々の仮想機能４３０、４４０、４５０は完全な仮想ホストチャネルアダプタ（virtual Host Channel Adapt

er：ｖＨＣＡ）であり、これは、ハードウェアにおいて、ＶＦに割当てられたＶＭに、ＩＢアドレス一式（たとえばＧＩＤ、ＧＵＩＤ、ＬＩＤ）および専用のＱＰスペースが割当てられていることを意味する。残りのネットワークおよびＳＭについては、ＨＣＡ４００は、仮想スイッチ４１５を介して追加のノードが接続されているスイッチのように見えている。ハイパーバイザ４１０はＰＦ４２０を用いることができ、（仮想機能に付与された）ＶＭはＶＦを用いる。

【００５８】

一実施形態に従うと、ｖＳｗｉｔｃｈアーキテクチャは、トランスペアレントな仮想化を提供する。しかしながら、各々の仮想機能には固有のＬＩＤが割当てられているので、利用可能な数のＬＩＤが速やかに消費される。同様に、多くのＬＩＤアドレスが（すなわち、各々の物理機能および各々の仮想機能ごとに１つずつ）使用されている場合、より多くの通信経路をＳＭによって演算しなければならず、それらのＬＦＴを更新するために、より多くのサブネット管理パケット（ＳＭＰ）をスイッチに送信しなければならない。たとえば、通信経路の演算は大規模ネットワークにおいては数分かかる可能性がある。ＬＩＤスペースが４９１５１個のユニキャストＬＩＤに制限されており、（ＶＦを介する）各々のＶＭとして、物理ノードおよびスイッチがＬＩＤを１つずつ占有するので、ネットワークにおける物理ノードおよびスイッチの数によってアクティブなＶＭの数が制限されてしまい、逆の場合も同様に制限される。

【００５９】

インフィニバンドＳＲ－ＩＯＶアーキテクチャモデル－仮想ポート（ｖＰｏｒｔ）

図６は、一実施形態に従った例示的なｖＰｏｒｔの概念を示す。図に示されるように、ホスト３００（たとえばホストチャネルアダプタ）は、さまざまな仮想機能３３０、３４０および３５０をいくつかの仮想マシンに割当てることができるハイパーバイザ４１０と対話することができる。同様に、物理機能はハイパーバイザ３１０によって処理することができる。

【００６０】

一実施形態に従うと、ベンダーに実装の自由を与えるためにｖＰｏｒｔ概念は緩やかに定義されており（たとえば、当該定義では、実装がＳＲＩＯＶ専用とすべきであるとは規定されていない）、ｖＰｏｒｔの目的は、ＶＭがサブネットにおいて処理される方法を標準化することである。ｖＰｏｒｔ概念であれば、空間ドメインおよび性能ドメインの両方においてよりスケーラブルであり得る、ＳＲ－ＩＯＶ共有のポートのようなアーキテクチャおよびｖＳｗｉｔｃｈのようなアーキテクチャの両方、または、これらのアーキテクチャの組合せが規定され得る。また、ｖＰｏｒｔはオプションのＬＩＤをサポートするとともに、共有のポートとは異なり、ＳＭは、ｖＰｏｒｔが専用のＬＩＤを用いていなくても、サブネットにおいて利用可能なすべてのｖＰｏｒｔを認識する。

【００６１】

インフィニバンドＳＲ－ＩＯＶアーキテクチャモデル－ＬＩＤが予めポピュレートされたｖＳｗｉｔｃｈ

一実施形態に従うと、本開示は、ＬＩＤが予めポピュレートされたｖＳｗｉｔｃｈアーキテクチャを提供するためのシステムおよび方法を提供する。

【００６２】

図７は、一実施形態に従った、ＬＩＤが予めポピュレートされた例示的なｖＳｗｉｔｃｈアーキテクチャを示す。図に示されるように、いくつかのスイッチ５０１～５０４は、ネットワーク切替環境６００（たとえばＩＢサブネット）内においてインフィニバンドファブリックなどのファブリックのメンバ間で通信を確立することができる。ファブリックはホストチャネルアダプタ５１０、５２０、５３０などのいくつかのハードウェアデバイスを含み得る。さらに、ホストチャネルアダプタ５１０、５２０および５３０は、それぞれ、ハイパーバイザ５１１、５２１および５３１と対話することができる。各々のハイパーバイザは、さらに、ホストチャネルアダプタと共に、いくつかの仮想機能５１４、５１５、５１６、５２４、５２５、５２６、５３４、５３５および５３６と対話し、設定し、

いくつかの仮想マシンに割当てることができる。たとえば、仮想マシン１　５５０はハイパーバイザ５１１によって仮想機能１　５１４に割当てることができる。ハイパーバイザ５１１は、加えて、仮想マシン２　５５１を仮想機能２　５１５に割当て、仮想マシン３　５５２を仮想機能３　５１６に割当てることができる。ハイパーバイザ５３１は、さらに、仮想マシン４　５５３を仮想機能１　５３４に割当てることができる。ハイパーバイザは、ホストチャネルアダプタの各々の上で十分な機能を有する物理機能５１３、５２３および５３３を介してホストチャネルアダプタにアクセスすることができる。

【００６３】

一実施形態に従うと、スイッチ５０１～５０４の各々はいくつかのポート（図示せず）を含み得る。いくつかのポートは、ネットワーク切替環境６００内においてトラフィックを方向付けるためにリニアフォワーディングテーブルを設定するのに用いられる。

【００６４】

一実施形態に従うと、仮想スイッチ５１２、５２２および５３２は、それぞれのハイパーバイザ５１１、５２１、５３１によって処理することができる。このようなｖＳｗｉｔｃｈアーキテクチャにおいては、各々の仮想機能は完全な仮想ホストチャネルアダプタ（ｖＨＣＡ）であり、これは、ハードウェアにおいて、ＶＦに割当てられたＶＭに、ＩＢアドレス一式（たとえばＧＩＤ、ＧＵＩＤ、ＬＩＤ）および専用のＱＰスペースが割当てられていることを意味する。残りのネットワークおよびＳＭ（図示せず）については、ＨＣＡ５１０、５２０および５３０は、仮想スイッチを介して追加のノードが接続されているスイッチのように見えている。

【００６５】

一実施形態に従うと、本開示は、ＬＩＤが予めポピュレートされたｖＳｗｉｔｃｈアーキテクチャを提供するためのシステムおよび方法を提供する。図７を参照すると、ＬＩＤは、さまざまな物理機能５１３、５２３および５３３に、さらには、仮想機能５１４～５１６、５２４～５２６、５３４～５３６（その時点でアクティブな仮想マシンに関連付けられていない仮想機能であっても）にも、予めポピュレートされている。たとえば、物理機能５１３はＬＩＤ１が予めポピュレートされており、仮想機能１　５３４はＬＩＤ１０が予めポピュレートされている。ネットワークがブートされているとき、ＬＩＤはＳＲ－ＩＯＶ　ｖＳｗｉｔｃｈ対応のサブネットにおいて予めポピュレートされている。ＶＦのすべてがネットワークにおけるＶＭによって占有されていない場合であっても、ポピュレートされたＶＦには、図７に示されるようにＬＩＤが割当てられている。

【００６６】

一実施形態に従うと、多くの同様の物理的なホストチャネルアダプタが２つ以上のポートを有することができ（冗長性のために２つのポートが共用となっている）、仮想ＨＣＡも２つのポートで表わされ、１つまたは２つ以上の仮想スイッチを介して外部ＩＢサブネットに接続され得る。

【００６７】

一実施形態に従うと、ＬＩＤが予めポピュレートされたｖＳｗｉｔｃｈアーキテクチャにおいては、各々のハイパーバイザは、それ自体のための１つのＬＩＤをＰＦを介して消費し、各々の追加のＶＦごとに１つ以上のＬＩＤを消費することができる。ＩＢサブネットにおけるすべてのハイパーバイザにおいて利用可能なすべてのＶＦを合計すると、サブネットにおいて実行することが可能なＶＭの最大量が得られる。たとえば、サブネット内の１ハイパーバイザごとに１６個の仮想機能を備えたＩＢサブネットにおいては、各々のハイパーバイザは、サブネットにおいて１７個のＬＩＤ（１６個の仮想機能ごとに１つのＬＩＤと、物理機能のために１つのＬＩＤ）を消費する。このようなＩＢサブネットにおいては、単一のサブネットについて理論上のハイパーバイザ限度は利用可能なユニキャストＬＩＤの数によって規定されており、（４９１５１個の利用可能なＬＩＤをハイパーバイザごとに１７個のＬＩＤで割って得られる）２８９１であり、ＶＭの総数（すなわち限度）は（ハイパーバイザごとに２８９１個のハイパーバイザに１６のＶＦを掛けて得られる）４６２５６である（実質的には、ＩＢサブネットにおける各々のスイッチ、ルータま

たは専用のＳＭノードが同様にＬＩＤを消費するので、これらの数は実際にはより小さくなる）。なお、ｖＳｗｉｔｃｈが、ＬＩＤをＰＦと共有することができるので、付加的なＬＩＤを占有する必要がないことに留意されたい。

【００６８】

一実施形態に従うと、ＬＩＤが予めポピュレートされたｖＳｗｉｔｃｈアーキテクチャにおいては、ネットワークが一旦ブートされると、すべてのＬＩＤについて通信経路が計算される。新しいＶＭを始動させる必要がある場合、システムは、サブネットにおいて新しいＬＩＤを追加する必要はない。それ以外の場合、経路の再計算を含め、ネットワークを完全に再構成させ得る動作は、最も時間を消費する要素となる。代わりに、ＶＭのための利用可能なポートはハイパーバイザのうちの１つに位置し（すなわち利用可能な仮想機能）、仮想マシンは利用可能な仮想機能に付与されている。

【００６９】

一実施形態に従うと、ＬＩＤが予めポピュレートされたｖＳｗｉｔｃｈアーキテクチャはまた、同じハイパーバイザによってホストされているさまざまなＶＭに達するために、さまざまな経路を計算して用いる能力を可能にする。本質的には、これは、ＬＩＤを連続的にすることを必要とするＬＭＣの制約によって拘束されることなく、１つの物理的なマシンに向かう代替的な経路を設けるために、このようなサブネットおよびネットワークがＬＩＤマスク制御ライク（LID-Mask-Control-like：ＬＭＣライク）な特徴を用いることを可能にする。ＶＭをマイグレートしてその関連するＬＩＤを宛先に送達する必要がある場合、不連続なＬＩＤを自由に使用できることは特に有用となる。

【００７０】

一実施形態に従うと、ＬＩＤが予めポピュレートされたｖＳｗｉｔｃｈアーキテクチャについての上述の利点と共に、いくつかの検討事項を考慮に入れることができる。たとえば、ネットワークがブートされているときに、ＳＲ－ＩＯＶ　ｖＳｗｉｔｃｈ対応のサブネットにおいてＬＩＤが予めポピュレートされているので、（たとえば起動時の）最初の経路演算はＬＩＤが予めポピュレートされていなかった場合よりも時間が長くかかる可能性がある。

【００７１】

インフィニバンドＳＲ－ＩＯＶアーキテクチャモデル－動的ＬＩＤ割当てがなされたｖＳｗｉｔｃｈ

一実施形態に従うと、本開示は、動的ＬＩＤ割当てがなされたｖＳｗｉｔｃｈアーキテクチャを提供するためのシステムおよび方法を提供する。

【００７２】

図８は、一実施形態に従った、動的ＬＩＤ割当てがなされた例示的なｖＳｗｉｔｃｈアーキテクチャを示す。図に示されるように、いくつかのスイッチ５０１～５０４は、ネットワーク切替環境７００（たとえばＩＢサブネット）内においてインフィニバンドファブリックなどのファブリックのメンバ間で通信を確立することができる。ファブリックは、ホストチャネルアダプタ５１０、５２０、５３０などのいくつかのハードウェアデバイスを含み得る。ホストチャネルアダプタ５１０、５２０および５３０は、さらに、ハイパーバイザ５１１、５２１および５３１とそれぞれ対話することができる。各々のハイパーバイザは、さらに、ホストチャネルアダプタと共に、いくつかの仮想機能５１４、５１５、５１６、５２４、５２５、５２６、５３４、５３５および５３６と対話し、設定し、いくつかの仮想マシンに割当てることができる。たとえば、仮想マシン１　５５０はハイパーバイザ５１１によって仮想機能１　５１４に割当てることができる。ハイパーバイザ５１１は、加えて、仮想マシン２　５５１を仮想機能２　５１５に割当て、仮想マシン３　５５２を仮想機能３　５１６に割当てることができる。ハイパーバイザ５３１はさらに、仮想マシン４　５５３を仮想機能１　５３４に割当てることができる。ハイパーバイザは、ホストチャネルアダプタの各々の上において十分な機能を有する物理機能５１３、５２３および５３３を介してホストチャネルアダプタにアクセスすることができる。

【００７３】

一実施形態に従うと、スイッチ５０１～５０４の各々はいくつかのポート（図示せず）を含み得る。いくつかのポートは、ネットワーク切替環境７００内においてトラフィックを方向付けるためにリニアフォワーディングテーブルを設定するのに用いられる。

【００７４】

一実施形態に従うと、仮想スイッチ５１２、５２２および５３２は、それぞれのハイパーバイザ５１１、５２１および５３１によって処理することができる。このようなｖＳｗｉｔｃｈアーキテクチャにおいては、各々の仮想機能は完全な仮想ホストチャネルアダプタ（ｖＨＣＡ）であり、これは、ハードウェアにおいて、ＶＦに割当てられたＶＭに、ＩＢアドレス一式（たとえばＧＩＤ、ＧＵＩＤ、ＬＩＤ）および専用のＱＰスペースが割当てられていることを意味する。残りのネットワークおよびＳＭ（図示せず）については、ＨＣＡ５１０、５２０および５３０は、仮想スイッチを介して、追加のノードが接続されているスイッチのように見えている。

【００７５】

一実施形態に従うと、本開示は、動的ＬＩＤ割当てがなされたｖＳｗｉｔｃｈアーキテクチャを提供するためのシステムおよび方法を提供する。図８を参照すると、ＬＩＤには、さまざまな物理機能５１３、５２３および５３３が動的に割当てられており、物理機能５１３がＬＩＤ１を受取り、物理機能５２３がＬＩＤ２を受取り、物理機能５３３がＬＩＤ３を受取る。アクティブな仮想マシンに関連付けられたそれらの仮想機能はまた、動的に割当てられたＬＩＤを受取ることもできる。たとえば、仮想マシン１　５５０がアクティブであり、仮想機能１　５１４に関連付けられているので、仮想機能５１４にはＬＩＤ５が割当てられ得る。同様に、仮想機能２　５１５、仮想機能３　５１６および仮想機能１　５３４は、各々、アクティブな仮想機能に関連付けられている。このため、これらの仮想機能にＬＩＤが割当てられ、ＬＩＤ７が仮想機能２　５１５に割当てられ、ＬＩＤ１１が仮想機能３　５１６に割当てられ、ＬＩＤ９が仮想機能１　５３４に割当てられている。ＬＩＤが予めポピュレートされたｖＳｗｉｔｃｈとは異なり、アクティブな仮想マシンにその時点で関連付けられていない仮想機能はＬＩＤの割当てを受けない。

【００７６】

一実施形態に従うと、動的ＬＩＤ割当てがなされていれば、最初の経路演算を実質的に減らすことができる。ネットワークが初めてブートしており、ＶＭが存在していない場合、比較的少数のＬＩＤを最初の経路計算およびＬＦＴ分配のために用いることができる。

【００７７】

一実施形態に従うと、多くの同様の物理的なホストチャネルアダプタが２つ以上のポートを有することができ（冗長性のために２つのポートが共用となっている）、仮想ＨＣＡも２つのポートで表わされ、１つまたは２つ以上の仮想スイッチを介して外部ＩＢサブネットに接続され得る。

【００７８】

一実施形態に従うと、動的ＬＩＤ割当てがなされたｖＳｗｉｔｃｈを利用するシステムにおいて新しいＶＭが作成される場合、どのハイパーバイザ上で新しく追加されたＶＭをブートすべきであるかを決定するために、自由なＶＭスロットが発見され、固有の未使用のユニキャストＬＩＤも同様に発見される。しかしながら、新しく追加されたＬＩＤを処理するためのスイッチのＬＦＴおよびネットワークに既知の経路が存在しない。新しく追加されたＶＭを処理するために新しいセットの経路を演算することは、いくつかのＶＭが毎分ごとにブートされ得る動的な環境においては望ましくない。大規模なＩＢサブネットにおいては、新しい１セットのルートの演算には数分かかる可能性があり、この手順は、新しいＶＭがブートされるたびに繰返されなければならないだろう。

【００７９】

有利には、一実施形態に従うと、ハイパーバイザにおけるすべてのＶＦがＰＦと同じアップリンクを共有しているので、新しいセットのルートを演算する必要はない。ネットワークにおけるすべての物理スイッチのＬＦＴを繰返し、（ＶＭが作成されている）ハイパーバイザのＰＦに属するＬＩＤエントリから新しく追加されたＬＩＤにフォワーディング

ポートをコピーし、かつ、特定のスイッチの対応するＬＦＴブロックを更新するために単一のＳＭＰを送信するだけでよい。これにより、当該システムおよび方法では、新しいセットのルートを演算する必要がなくなる。

【００８０】

一実施形態に従うと、動的ＬＩＤ割当てアーキテクチャを備えたｖＳｗｉｔｃｈにおいて割当てられたＬＩＤは連続的である必要はない。各々のハイパーバイザ上のＶＭ上で割当てられたＬＩＤをＬＩＤが予めポピュレートされたｖＳｗｉｔｃｈと動的ＬＩＤ割当てがなされたｖＳｗｉｔｃｈとで比較すると、動的ＬＩＤ割当てアーキテクチャにおいて割当てられたＬＩＤが不連続であり、そこに予めポピュレートされたＬＩＤが本質的に連続的であることが分かるだろう。さらに、ｖＳｗｉｔｃｈ動的ＬＩＤ割当てアーキテクチャにおいては、新しいＶＭが作成されると、次に利用可能なＬＩＤが、ＶＭの生存期間の間中ずっと用いられる。逆に、ＬＩＤが予めポピュレートされたｖＳｗｉｔｃｈにおいては、各々のＶＭは、対応するＶＦに既に割当てられているＬＩＤを引継ぎ、ライブマイグレーションのないネットワークにおいては、所与のＶＦに連続的に付与されたＶＭが同じＬＩＤを得る。

【００８１】

一実施形態に従うと、動的ＬＩＤ割当てアーキテクチャを備えたｖＳｗｉｔｃｈは、いくらかの追加のネットワークおよびランタイムＳＭオーバーヘッドを犠牲にして、予めポピュレートされたＬＩＤアーキテクチャモデルを備えたｖＳｗｉｔｃｈの欠点を解決することができる。ＶＭが作成されるたびに、作成されたＶＭに関連付けられた、新しく追加されたＬＩＤで、サブネットにおける物理スイッチのＬＦＴが更新される。この動作のために、１スイッチごとに１つのサブネット管理パケット（ＳＭＰ）が送信される必要がある。各々のＶＭがそのホストハイパーバイザと同じ経路を用いているので、ＬＭＣのような機能も利用できなくなる。しかしながら、すべてのハイパーバイザに存在するＶＦの合計に対する制限はなく、ＶＦの数は、ユニキャストＬＩＤの限度を上回る可能性もある。このような場合、当然、アクティブなＶＭ上でＶＦのすべてが必ずしも同時に付与されることが可能になるわけではなく、より多くの予備のハイパーバイザおよびＶＦを備えることにより、ユニキャストＬＩＤ限度付近で動作する際に、断片化されたネットワークの障害を回復および最適化させるための融通性が追加される。

【００８２】

インフィニバンドＳＲ－ＩＯＶアーキテクチャモデル－動的ＬＩＤ割当てがなされかつＬＩＤが予めポピュレートされたｖＳｗｉｔｃｈ

図９は、一実施形態に従った、動的ＬＩＤ割当てがなされてＬＩＤが予めポピュレートされたｖＳｗｉｔｃｈを備えた例示的なｖＳｗｉｔｃｈアーキテクチャを示す。図に示されるように、いくつかのスイッチ５０１～５０４は、ネットワーク切替環境８００（たとえばＩＢサブネット）内においてインフィニバンドファブリックなどのファブリックのメンバ間で通信を確立することができる。ファブリックはホストチャネルアダプタ５１０、５２０、５３０などのいくつかのハードウェアデバイスを含み得る。ホストチャネルアダプタ５１０、５２０および５３０は、それぞれ、さらに、ハイパーバイザ５１１、５２１および５３１と対話することができる。各々のハイパーバイザは、さらに、ホストチャネルアダプタと共に、いくつかの仮想機能５１４、５１５、５１６、５２４、５２５、５２６、５３４、５３５および５３６と対話し、設定し、いくつかの仮想マシンに割当てることができる。たとえば、仮想マシン１　５５０は、ハイパーバイザ５１１によって仮想機能１　５１４に割当てることができる。ハイパーバイザ５１１は、加えて、仮想マシン２　５５１を仮想機能２　５１５に割当てることができる。ハイパーバイザ５２１は、仮想マシン３　５５２を仮想機能３　５２６に割当てることができる。ハイパーバイザ５３１は、さらに、仮想マシン４　５５３を仮想機能２　５３５に割当てることができる。ハイパーバイザは、ホストチャネルアダプタの各々の上において十分な機能を有する物理機能５１３、５２３および５３３を介してホストチャネルアダプタにアクセスすることができる。

【００８３】
一実施形態に従うと、スイッチ５０１～５０４の各々はいくつかのポート（図示せず）を含み得る。これらいくつかのポートは、ネットワーク切替環境８００内においてトラフィックを方向付けるためにリニアフォワーディングテーブルを設定するのに用いられる。
【００８４】
一実施形態に従うと、仮想スイッチ５１２、５２２および５３２は、それぞれのハイパーバイザ５１１、５２１、５３１によって処理することができる。このようなｖＳｗｉｔｃｈアーキテクチャにおいては、各々の仮想機能は、完全な仮想ホストチャネルアダプタ（ｖＨＣＡ）であり、これは、ハードウェアにおいて、ＶＦに割当てられたＶＭに、ＩＢアドレス一式（たとえばＧＩＤ、ＧＵＩＤ、ＬＩＤ）および専用のＱＰスペースが割当てられていることを意味する。残りのネットワークおよびＳＭ（図示せず）については、ＨＣＡ５１０、５２０および５３０は、仮想スイッチを介して、追加のノードが接続されているスイッチのように見えている。
【００８５】
一実施形態に従うと、本開示は、動的ＬＩＤ割当てがなされＬＩＤが予めポピュレートされたハイブリッドｖＳｗｉｔｃｈアーキテクチャを提供するためのシステムおよび方法を提供する。図９を参照すると、ハイパーバイザ５１１には、予めポピュレートされたＬＩＤアーキテクチャを備えたｖＳｗｉｔｃｈが配置され得るとともに、ハイパーバイザ５２１には、ＬＩＤが予めポピュレートされて動的ＬＩＤ割当てがなされたｖＳｗｉｔｃｈが配置され得る。ハイパーバイザ５３１には、動的ＬＩＤ割当てがなされたｖＳｗｉｔｃｈが配置され得る。このため、物理機能５１３および仮想機能５１４～５１６には、それらのＬＩＤが予めポピュレートされている（すなわち、アクティブな仮想マシンに付与されていない仮想機能であってもＬＩＤが割当てられている）。物理機能５２３および仮想機能１　５２４にはそれらのＬＩＤが予めポピュレートされ得るとともに、仮想機能２　５２５および仮想機能３　５２６にはそれらのＬＩＤが動的に割当てられている（すなわち、仮想機能２　５２５は動的ＬＩＤ割当てのために利用可能であり、仮想機能３　５２６は、仮想マシン３　５５２が付与されているので、１１というＬＩＤが動的に割当てられている）。最後に、ハイパーバイザ３　５３１に関連付けられた機能（物理機能および仮想機能）にはそれらのＬＩＤを動的に割当てることができる。これにより、結果として、仮想機能１　５３４および仮想機能３　５３６が動的ＬＩＤ割当てのために利用可能となるとともに、仮想機能２　５３５には、仮想マシン４　５５３が付与されているので、９というＬＩＤが動的に割当てられている。
【００８６】
ＬＩＤが予めポピュレートされたｖＳｗｉｔｃｈおよび動的ＬＩＤ割当てがなされたｖＳｗｉｔｃｈがともに（いずれかの所与のハイパーバイザ内で独立して、または組合わされて）利用されている、図９に示されるような一実施形態に従うと、ホストチャネルアダプタごとの予めポピュレートされたＬＩＤの数はファブリックアドミニストレータによって定義することができ、（ホストチャネルアダプタごとに）０＜＝予めポピュレートされたＶＦ＜＝総ＶＦの範囲内になり得る。動的ＬＩＤ割当てのために利用可能なＶＦは、（ホストチャネルアダプタごとに）ＶＦの総数から予めポピュレートされたＶＦの数を減じることによって見出すことができる。
【００８７】
一実施形態に従うと、多くの同様の物理的なホストチャネルアダプタが２つ以上のポートを有することができ（冗長性のために２つのポートが共用となっている）、仮想ＨＣＡも２つのポートで表わされ、１つまたは２つ以上の仮想スイッチを介して外部ＩＢサブネットに接続され得る。
【００８８】
インフィニバンド－サブネット間通信（ファブリックマネージャ）
一実施形態によれば、本開示の実施形態は、単一サブネット内にインフィニバンドファブリックを提供することに加えて、２つ以上のサブネットにまたがるインフィニバンドフ

ァブリックを提供することもできる。
【００８９】
図１０は、一実施形態による例示的なマルチサブネットインフィニバンドファブリックを示す。図に示すように、サブネットＡ１０００内では、ある数のスイッチ１００１～１００４が、サブネットＡ１０００（例えば、ＩＢサブネット）内において、インフィニバンドファブリックなどのファブリックのメンバ間において通信を提供することができる。ファブリックは、例えば、チャネルアダプタ１０１０などのある数のハードウェアデバイスを含むことができる。ホストチャネルアダプタ１０１０は、次いで、ハイパーバイザ１０１１と対話することができる。ハイパーバイザは、次いで、それが対話するホストチャネルアダプタと関連して、ある数の仮想機能１０１４をセットアップすることができる。ハイパーバイザは、加えて、仮想マシンを仮想機能の各々に割り当てることができ、仮想マシン１　１０１０５が仮想機能１　１０１４に割り当てられるなどする。ハイパーバイザは、各ホストチャネルアダプタ上において、物理機能１０１３など、十分な機能を有する物理機能を介して、それらの関連付けられるホストチャネルアダプタにアクセスできる。サブネットＢ１０４０内では、ある数のスイッチ１０２１～１０２４が、サブネットＢ１０４０（例えば、ＩＢサブネット）内において、インフィニバンドファブリックなどのファブリックのメンバ間において通信を提供することができる。ファブリックは、例えば、チャネルアダプタ１０３０などのある数のハードウェアデバイスを含むことができる。ホストチャネルアダプタ１０３０は、次いで、ハイパーバイザ１０３１と対話することができる。ハイパーバイザは、次いで、それが対話するホストチャネルアダプタと関連して、ある数の仮想機能１０３４をセットアップすることができる。ハイパーバイザは、加えて、仮想マシンを仮想機能の各々に割り当てることができ、仮想マシン２　１０３５が仮想機能２　１０３４に割り当てられるなどする。ハイパーバイザは、各ホストチャネルアダプタ上において、物理機能１０３３など、十分な機能を有する物理機能を介して、それらの関連付けられるホストチャネルアダプタにアクセスできる。各サブネット（すなわち、サブネットＡおよびサブネットＢ）内には１つのホストチャネルアダプタしか示されていないが、複数のホストチャネルアダプタおよびそれらの対応するコンポーネントを各サブネット内に含めることができることを理解されたい。
【００９０】
一実施形態によれば、ホストチャネルアダプタの各々は、仮想スイッチ１０１２および仮想スイッチ１０３２などの仮想スイッチにさらに関連付けることができ、各ＨＣＡは、上述したように、異なるアーキテクチャモデルでセットアップすることができる。図１０内の両方のサブネットは、事前にポピュレートされたＬＩＤアーキテクチャモデルを有するｖＳｗｉｔｃｈを使用しているように示されているが、これはすべてのそのようなサブネット構成が同様のアーキテクチャモデルに従わなければならないことを意味するものではない。
【００９１】
一実施形態によれば、各サブネット内の少なくとも１つのスイッチはルータに関連付けられることができ、サブネットＡ１０００内のスイッチ１００２はルータ１００５に関連付けられ、サブネットＢ１０４０内のスイッチ１０２１はルータ１００６に関連付けられるなどする。
【００９２】
一実施形態によれば、少なくとも１つのデバイス（例えば、スイッチ、ノードなど）をファブリックマネージャ（図示せず）に関連付けることができる。ファブリックマネージャは、例えば、サブネット間ファブリックトポロジを発見し、ファブリックプロファイル（例えば、仮想マシンファブリックプロファイル）を作成し、仮想マシンファブリックプロファイルを構築するための基礎を形成する仮想マシン関連データベースオブジェクトを構築するために使用することができる。加えて、ファブリックマネージャは、どのサブネットがどのパーティション番号を使用してどのルータポートを介して通信することが許可されているかに関して、法的なサブネット間接続を定義することができる。

【００９３】
一実施形態によれば、サブネットＡ内の仮想マシン１などの発信元でのトラフィックが、サブネットＢ内の仮想マシン２などの異なるサブネットの宛先にアドレス指定されている場合、トラフィックはサブネットＡ内のルータ、すなわちルータ１００５にアドレス指定され、ルータ１００５は次いでそのトラフィックをルータ１００６とのそれのリンクを介してサブネットＢに渡すことができる。
【００９４】
スケーラブルなビットマップに基づくＰ＿Ｋｅｙテーブル
一実施形態によれば、仮想マシンを使用を提供された拡張されたインフィニバンドネットワークのため、可能性のあるパーティションの数が劇的に増加している。しかしながら、Ｐ＿Ｋｅｙ管理およびルーティングを扱う現在の方法では、トラフィックがファブリックを通過する際にかなりのオーバヘッド時間が追加される。従来、インフィニバンド規格では、ＳＭが索引付けされたテーブルとしてアクセスできる１６ビットのＰ＿Ｋｅｙ値の配列として、Ｐ＿Ｋｅｙテーブルを定義している。この規格のハードウェア実現例は、ワイヤ速度パケットレートでＩＢパケットのパーティションチェックを実行するためにルックアップを実行するために連想メモリを使用することを意味する。実際には、これはハードウェアにより実現されるＰ＿Ｋｅｙテーブルの可能なサイズを、１６ビットのＰ＿Ｋｅｙ値が表す６４Ｋ値空間よりも桁違いに小さく制限する。
【００９５】
一実施形態によれば、インフィニバンド規格は、パーティションメンバシップを１６ビットＰ＿Ｋｅｙを介して定義し、ファブリック内の各ポートでＰ＿Ｋｅｙ値の固定テーブルを使用してパーティション分離を実施する。サブネットマネージャは、異なるテーブルエントリで異なるＰ＿Ｋｅｙ値をプログラミングできる。そのパーティション関連付けに基づいてＰ＿Ｋｅｙでそのヘッダにおいてマークされたパケットがポートに到着すると、下位のハードウェアは、入来するパケットを、チェックするハードウェアに関連付けられるＰ＿Ｋｅｙテーブル内のすべての値と比較することができる（すなわち、テーブルで連想ルックアップを実行する）。しかしながら、このルックアップは多数のパーティションではうまくスケーリングせず、テーブル内である数のパーティション上でルックアップを実行している間に不要なオーバヘッドが発生する可能性がある。
【００９６】
図１１は、一実施形態による、パーティション分離のための連想テーブルを有する例示的なインフィニバンドファブリックを示す。図１１に示すように、１つ以上のエンドノード１１４１～１１４４を、ネットワークファブリック１１００において接続することができる。ネットワークファブリック１１００は、複数のリーフスイッチ１１１１～１１１４と、複数のスパインスイッチまたはルートスイッチ１１３１～１１３４とを含む、ファットツリートポロジに基づくことができる。加えて、ネットワークファブリック１１００は、スイッチ１１２１～１１２４のような１つ以上の中間スイッチを含むことができる。また、図１１に示すように、エンドノード１１４１～１１４４の各々はマルチホームノードであってもよい。
【００９７】
一実施形態によれば、ノード１１４１～１１４４の各々は、ある数のパーティションのうちの１つ以上に属することができ、各パーティションは、あるＰ＿Ｋｅｙ値に関連付けられる。上述したように、Ｐ＿Ｋｅｙ値は、ファブリック内で送信される各パケットに含めることができ、サブネット内におけるパーティション実施の程度を変化させることができる。Ｐ＿Ｋｅｙ値の各々は、ある数のテーブル、例えばテーブル１１０１～１１０８（ここでは「ポートＰ＿Ｋｅｙテーブル」とも呼ばれる）に格納され、各テーブルは、例えば、スイッチのポート、またはホストノードに属するＨＣＡポートなど、ファブリックの他のメンバのポートに関連付けられる。各テーブル１１０１～１１０８は、サブネットマネージャによって設定することができる。
【００９８】

一実施形態によれば、各ポートＰ＿Ｋｅｙテーブルは、各エントリが１６ビットＰ＿Ｋｅｙ値を含むことができるＮ個のエントリの配列を含むことができ、Ｎの値は、（ポートのステータスおよび能力を発見する一環としてＳＭによって検索される）ポートの能力である。ポートがパーティションのメンバであるはずである（つまり、フルまたは制限付き）ことを意味するポリシー情報をＳＭがポートが持っている場合、ＳＭはポートＰ＿Ｋｅｙテーブルにおいて次に利用可能なエントリを見つけて、関連する１６ビットＰ＿Ｋｅｙ値をそのエントリに格納できる。Ｐ＿Ｋｅｙ値が成功裡に格納される（ＳＭＡによってＳＭに確認される）と、ＳＭは、ポートハードウェアがポートを通過するすべてのパケットについて新たなＰ＿Ｋｅｙテーブルエントリの一致を含むことを予期することができ、パケットＰ＿Ｋｅｙ値が更新されたＰ＿Ｋｅｙテーブルエントリにおける新たな値と一致するパケットを受け入れる（すなわち、ポートを通過させる）ことができる。

【００９９】

一実施形態によれば、サブネット内のポートがもはやパーティションのメンバではないと想定される場合、ＳＭは関連するＰ＿Ｋｅｙテーブルエントリをクリアする（０を書き込む）ことができる。Ｐ＿Ｋｅｙ値が成功裡にクリアされる（ＳＭＡによってＳＭに確認される）と、ＳＭは、ポートハードウェアが、丁度ポートＰ＿Ｋｅｙテーブルからクリアされたものに対応するパケットＰ＿Ｋｅｙ値を有する、そのポートを通過するすべてのパケットについてＰ＿Ｋｅｙ一致に失敗することを予期し得、次いでパケットをドロップし得る。

【０１００】

一実施形態によれば、ＳＭは、既存の値を最初にクリアすることなく、有効なＰ＿Ｋｅｙテーブルエントリを新たな値と置き換えることができる。この場合の期待される効果は、エントリの内容が最初にクリアされ、次いでその後新たな値で更新された場合と同じである。

【０１０１】

すでに説明したように、図１１に示す実施形態は、多数のパーティションを持たないサブネットの範囲内で十分にスケーリングすることができる。しかしながら、例えば、エンドノードおよび１つ以上のｖＳｗｉｔｃｈアーキテクチャの仮想化を利用するサブネットなど、任意の所与のサブネット内に多数のパーティションが存在すると、上述の連想テーブルルックアップはオーバヘッド時間の大幅な増加を招き得、ＩＢ規格によって規定された関連するリンク速度要件が満たされるべきである場合、ハードウェアで実現することが可能でさえないかもしれない。

【０１０２】

一実施形態によれば、ＩＢファブリック内の各ポートに関連付けられるＰ＿Ｋｅｙテーブルを利用する代わりに、例示的な方法およびシステムは、ビットマップを利用して、可能なすべてのＰ＿Ｋｅｙ値の表現、‐１６ビット（すなわち６４ｋの可能な値）をハードウェアにおいて実現できる。そのような方法およびシステムでは、ハードウェア実現例内において、各可能なＰ＿Ｋｅｙ値を単一ビットとして表現することができ、単一ビットの値（例えば、１または０）は、Ｐ＿Ｋｅｙ値が許可される（すなわち、パケットは、パーティションチェックを実行するポートを通過することを許可される）か、またはポートに関連付けられない（すなわち、ＩＢパケットは、次いで、ＩＢ規格によって規定された関連するポートタイプについてのＰ＿Ｋｅｙ実施規則に従ってそのパーティション番号に対する制限付きおよびフルメンバエントリの両方についてのビットマップエントリとパケットＰ＿Ｋｅｙを相関させることに基づいて、パーティション実施規則に従ってチェックされ得る）かを定義できる。

【０１０３】

図１２は、一実施形態による、パーティション分離のためのビットマップを有する例示的なインフィニバンドファブリックを示す。図１２に示すように、１つ以上のエンドノード１２４１～１２４４を、ネットワークファブリック１２００において接続することができる。ネットワークファブリック１２００は、複数のリーフスイッチ１２１１～１２１４

と、複数のスパインスイッチまたはルートスイッチ１２３１～１２３４とを含む、ファットツリートポロジに基づくことができる。加えて、ネットワークファブリック１２００は、スイッチ１２２１～１２２４のような１つ以上の中間スイッチを含むことができる。また、図１２に示すように、エンドノード１２４１～１２４４の各々はマルチホームノードであってもよい。

【０１０４】

一実施形態によれば、ノード１２４１～１２４４の各々は、ある数のパーティションのうちの１つ以上に属することができ、各パーティションは、あるＰ＿Ｋｅｙ値に関連付けられる。上述したように、Ｐ＿Ｋｅｙ値は、ファブリック内で送信される各パケットに含めることができ、サブネット内におけるパーティション実施の程度を変化させることができる。各Ｐ＿Ｋｅｙ値は、ビットマップ１２０１～１２０８のようなある数のビットマップで参照することができ、各ビットマップは、例えば、スイッチのポート、またはホストノードに属する（関連付けられる）ＨＣＡポートなど、ファブリックの他のメンバのポートに関連付けられる。各ビットマップ１２０１～１２０８は、サブネットマネージャによって設定することができる。

【０１０５】

一実施形態によれば、Ｐ＿Ｋｅｙを連想テーブルルックアップの代わりにビットマップで表すことにより、これにより、柔軟性が増大し、ファブリックにおけるパーティション分離およびパケットルーティングのオーバヘッド時間が短縮できる。どのノードも任意のパーティションのメンバになることができる。これにより、任意のスイッチポートが、ＳＭが構成したＰ＿Ｋｅｙだけをそのスイッチポートを介して許可されるようにすることができる（そうでない場合、パケットはドロップされ得る）。ビットマップに基づくＰ＿Ｋｅｙテーブルを利用することにより、方法およびシステムはほんのわずかな数のパーティションまたは数百のものを許容することができる。そのような方法およびシステムは、スイッチを通過することになっているパーティションのみを許可することを確実にすることができる。このような方法およびシステムは、より多くの数のパーティションが利用可能であっても、より大きなパーティション分離を可能にすることができる。

【０１０６】

一実施形態によれば、ビットマップスキームを使用することにより、Ｐ＿Ｋｅｙ値の連想ルックアップを有する代わりに、ハードウェアはパケットヘッダからＰ＿Ｋｅｙ値を取ることができ、連想ルックアップ機構に値を送る代わりに、Ｐ＿Ｋｅｙ値を取り、それを（例えば、６４Ｋビットの）ビット配列に索引付する。次いで、ＩＢパケットが、ＩＢ規格によって規定された関連するポートタイプについてのＰ＿Ｋｅｙ実施規則に従ってそのパーティション番号に対する制限付きおよびフルメンバエントリの両方についてのビットマップエントリとパケットＰ＿Ｋｅｙを相関させることに基づいて、パーティション実施規則に従ってチェックされ得る。これは、例えば、スケーリングの増加（すなわち、パフォーマンスの損失なしにより多数のパーティションＰ＿Ｋｅｙを可能にすること）に至り得る。

【０１０７】

一実施形態によれば、上述のビットマップスキームは、エンドノードでパーティション分離を実施することに加えて、エンドノードに向けて接続されるポートに対する第１のリーフスイッチ、第１のリーフスイッチとルートスイッチとの間の任意のスイッチ、および最後のリーフスイッチなど、中間スイッチでパーティション実施を追加的にサポートすることができる。スイッチからスイッチへのリンクが、ファブリックのルーティングに応じて、多くの異なるノードによって使用される。従来のシステムでは、パーティション実施は主にエンドノード上で使用され、エンドノードに直接接続されるポートに対する最後のリーフスイッチで実施されたかも知れないが、リーフスイッチよりも高いレベルでのスイッチでは実施されない（つまり、パーティション実施はスイッチ間のリンク上では使用されなかった）。しかしながら、許可されたＰ＿Ｋｅｙおよび許可されていないＰ＿Ｋｅｙに関連付けられるパーティションビットマップを使用することで得られる速度および効率

により、パーティション実施がこれらの中間リンクで起こり得る。各スイッチポートは許可されているパーティションがそのポートを介して送信されることを許すにすぎないことを確実にすることによって、中間リンクで通過できないポートに遭遇するパケットは、（エンドノードまたはリーフスイッチにおける代わりに）はるかによりすぐにドロップされ得、最終的にドロップされるパケットに対するリンク帯域幅使用率に関してパフォーマンスの低下を少なくすることができる。

【０１０８】

一実施形態によれば、ＳＭは、パーティションに応じて、完全なファブリックのどこでどのトラフィックがルーティングされることが許可されるかを、ビットマップで判断して設定することができる。ＳＭは、次いで、ビットマップを使用してパーティション実施をセットアップできる。ＳＭは、どのエンドノードがどのパーティションのメンバであるかに関するポリシー構成を受信することができる。このポリシー情報に基づいて、（ビットマップを介して）ＳＭはエンドノード上でパーティション実施を設定できる。ＳＭは、接続を切り替えるようにスイッチを構成することもできる。これを行った後、ＳＭはビットマップを使用してファブリック内のトラフィックフローを制限できる。ＳＭは、ビットマップ内のパーティション毎の値を１または０、パスまたはドロップ（すなわち、許可する、または許可しない）にセットする。これにより、許可されていないトラフィックを、宛先ノードまたは宛先ノードの前の最後のリーフスイッチにおいてではなく、トラフィックのルーティングにおいて早期にドロップできる。

【０１０９】

一実施形態によれば、各ビットマップは、各ポートに対するＰ＿Ｋｅｙ実施のための６４Ｋビット配列を含むビット配列とすることができる。このようなビットマップは、任意の種類のＩＢポート：スイッチ外部ポート、スイッチ管理ポート（すなわち、スイッチポート０）、ＴＣＡポート（例えば、ＩＢイーサネット（登録商標）ブリッジ／ゲートウェイノード実装用）、ＨＣＡポートおよびルータポートに関連付けられ得る。次いで、ＩＢパケットを、ＩＢ規格によって規定された関連するポートタイプについてのＰ＿Ｋｅｙ実施規則に従ってそのパーティション番号に対する制限付きおよびフルメンバエントリの両方についてのビットマップエントリとパケットＰ＿Ｋｅｙを相関させることに基づいて、パーティション実施規則に従ってチェックし得る。

【０１１０】

一実施形態によれば、単一のパーティション番号を２つのビット配列索引に関連付けることができ、１つは制限付きメンバシップを表すものであり、１つはフルメンバシップを表すものである。したがって、ポートがフルメンバシップのみを表すか、フルおよび制限付きメンバシップの両方を表すと考えられるかに応じて、フルメンバ索引のみまたはフルおよび制限付きメンバ索引の両方を設定して、対応するメンバシップトラフィックを許可する必要がある。しかしながら、制限付きパーティションメンバが別の制限付きメンバと通信することが許可されていない一方で、フルメンバがフルメンバおよび制限付きメンバの両方と通信することが許可されているという事実のため、ルックアップ実装はこれを考慮し、パケットを開始しているエンドノードが、制限付きメンバのパケットを、それがパーティションの制限付きメンバであるにすぎない場合に送信することが許可されるにすぎず、しかしながら、それは、対応して、フルメンバパケットを受信することが許可されるにすぎないことになるようにしなければならない。

【０１１１】

図１３は、一実施形態による、高性能コンピューティング環境においてスケーラブルなビットマップに基づくＰ＿Ｋｅｙテーブルをサポートするための方法のフローチャートである。

【０１１２】

ステップ１３１０において、パケットは、送信元（例えば、送信元ノードまたは送信元仮想マシン）から発信され、宛先（例えば、宛先ノードまたは宛先仮想マシン）に宛てられ得る。パケットは、他の情報の中でもとりわけ、送信元と宛先との間の共通パーティシ

ョンなどのパーティション情報に対応するＰ＿Ｋｅｙ値を含むことができるヘッダを含むことができる。

【０１１３】

ステップ１３２０において、パケットは、算出されたルートに沿ってインフィニバンドファブリックのようなファブリックを横断することを開始できる。ルートは、例えば、サブネットマネージャによって算出することができる。

【０１１４】

ステップ１３３０において、パケットは、ファブリック内のルーティングされた経路に沿ってスイッチのアドレス指定されたポートに到達することができる。

【０１１５】

ステップ１３４０で、アドレス指定されたポートが位置するスイッチは、パケットＰ＿Ｋｅｙをビットマップの内容と照合することができる（すなわち、ＩＢパケットは、ＩＢ規格によって規定された関連するポートタイプについてのＰ＿Ｋｅｙ実施規則に従ってそのパーティション番号に対する制限付きおよびフルメンバエントリの両方についてのビットマップエントリとパケットＰ＿Ｋｅｙを相関させることに基づいて、パーティション実施規則に従ってチェックされ得る）。

【０１１６】

ステップ１３５０で、チェックの結果が「許可」である場合、これは、パケットがルーティングされた経路に沿って宛先へ（ＤＬＩＤ）進むことを許可する。

【０１１７】

ステップ１３６０で、チェックの結果が「不許可」である場合、スイッチはアドレス指定されたポートでパケットをドロップし、パケットがルーティングされた経路に沿って進まないようにすることができる。

【０１１８】

図１３に示され、上述された実施形態は、ファブリック内の１つのスイッチのみでのパケットＰ＿Ｋｅｙのチェックを開示するが、ルーティングされた経路に沿った各スイッチおよびノードは、各アドレス指定されたポートで関連付けられるビットマップを使用してＰ＿Ｋｅｙのチェックを実行して、パケットが許可されているかいないかを判断することができることが理解されるべきである。

【０１１９】

ビットマップに基づくＨＷ実現例を使用する構成可能なレガシーＰ＿Ｋｅｙテーブル抽象化

一実施形態によれば、上記のビットマップに基づく実現例は、レガシーサブネットマネージャ実現例によって直接使用することができないという点で問題がある。これは、インフィニバンド規格（インフィニバンド（登録商標）トレード・アソシエーション・アーキテクチャ規格、第１巻、バージョン１．３（２０１５年３月リリース））で定義されているサブネットマネージャなどの、現在定義されている管理エンティティが、ビットマップに基づくＰ＿Ｋｅｙ実現例と対話するように定義されていないためであり、それらは、代わりに、ＩＢ規格によって定義されたレガシーＰ＿Ｋｅｙテーブルに基づくスキームと対話するように定義される。

【０１２０】

一実施形態によれば、ＤＲＡＭにおけるマッピングテーブルを、マッピングテーブルを実装するソフトウェアベースのＳＭＡの使用によって達成することができる。このマッピングテーブルを用いて、ビットマップに基づくＰ＿Ｋｅｙテーブルのレガシー準拠のビューを提供することが可能である。そのようなレガシー準拠のビューは、仮想Ｐ＿Ｋｅｙテーブル、または構成可能なレガシーＰ＿Ｋｅｙテーブル抽象化と呼ぶことができる。

【０１２１】

一実施形態によれば、仮想Ｐ＿Ｋｅｙテーブルのサイズは、その目的のために割り当てられ得るＤＲＡＭの量によって制限される。特定の実施形態では、アクセス可能なＤＲＡＭの量は、仮想Ｐ＿Ｋｅｙテーブルに関連付けられるスイッチまたは他のデバイス（エン

ドノードなど）内に含まれる。ある実施形態では、ＤＲＡＭの外部ソースを利用して、より大きな仮想Ｐ＿Ｋｅｙテーブルを管理エンティティによってアクセス可能にすることができる。

【０１２２】

一実施形態によれば、各仮想Ｐ＿Ｋｅｙテーブルは、構成パラメータ（またはヘッダファイル定数）で定義されたサイズを有することができ、関連するＳＭＡ属性インスタンスに関連付けられる対応するメモリデータ構造（例えば、埋め込みプロセッササブシステムメモリデータ構造（ＥＰＳ））によって実現することができる。

【０１２３】

一実施形態によれば、管理エンティティ（例えば、サブネットマネージャ）が現在クリアされている（値ゼロを含む）（仮想の）Ｐ＿ＫｅｙテーブルエントリにＰ＿Ｋｅｙ値を格納する度毎に、ＳＭＡ実現例はＨＷアクセスを呼び出して、Ｐ＿Ｋｅｙ値によって索引付けされたＨＷビット配列を１にセットでき、次いで、仮想Ｐ＿Ｋｅｙテーブルエントリ（例えば、ＥＰＳメモリデータ構造）を更新して、ＳＭによって指定されるＰ＿Ｋｅｙテーブルエントリ番号が指定されたＰ＿Ｋｅｙ値を含むようにすることができる。

【０１２４】

一実施形態によれば、ＳＭが仮想Ｐ＿ＫｅｙテーブルにおけるエントリからＰ＿Ｋｅｙ値をクリアする度毎に、ＳＭＡ実現例はメモリデータ構造からの古いＰ＿Ｋｅｙ値を使用し、次いで、仮想Ｐ＿Ｋｅｙテーブルエントリ（すなわち、ＥＰＳメモリデータ構造）もクリアされる前に、対応するＨＷビット配列索引をＨＷアクセスインターフェイスを介してクリアすることができる。

【０１２５】

一実施形態によれば、ＳＭが、現在クリアされていない（仮想）Ｐ＿Ｋｅｙテーブルエントリに新たなＰ＿Ｋｅｙ値を書き込む度毎に、ＳＭＡ実現例はまずＥＰＳメモリデータ構造からの古いＰ＿Ｋｅｙ値を使用し、次いで、対応するＨＷビット配列索引をＨＷアクセスインターフェイスを介してクリアすることができる。古い値がクリアされた後、ＳＭＡ実現例は、新たに追加されたＰ＿Ｋｅｙ値に対応するビット配列エントリを設定し、次いで、メモリデータ構造を更新して、指定されたＰ＿Ｋｅｙテーブルエントリが指定されたＰ＿Ｋｅｙ値を含むようにすることができる。

【０１２６】

図１４は、一実施形態による、パーティション分離のためのビットマップと、レガシー管理エンティティのためのＰ＿Ｋｅｙ抽象化とを有する例示的なファブリックを示す。図に示す実施形態では、スイッチ１４００は、ある数のポート１４２０を備えることができ、これらのポートの各々は、上述したように、Ｐ＿Ｋｅｙ実施のためのビットマップに基づくＰ＿Ｋｅｙテーブル１４１０に関連付けることができる。スイッチ１４００は、加えて、メモリ（例えばＤＲＡＭ）１４３０を備えることができる。

【０１２７】

一実施形態によれば、サブネットマネージャ１４４０（例えばレガシーサブネットマネージャ）は、ビットマップに基づくＰ＿Ｋｅｙテーブル１４１０を照会／設定することができないことがあり得る。このような状況において、スイッチのメモリ１４３０内にＰ＿Ｋｅｙテーブル抽象化１４５０（すなわち「仮想Ｐ＿Ｋｅｙテーブル」）を設けることができる。次いで、サブネットマネージャは、サブネットマネージャが連想メモリ／ルックアップに基づいてレガシーＰ＿Ｋｅｙテーブルと対話するであろうのと同じ態様で、仮想Ｐ＿Ｋｅｙテーブルを照会し、それと対話することができる。より具体的には、ＳＭ１４４０は、スイッチ１４００に関連付けられるＳＭＡ１４４１とインターフェイスすることができる。このインターフェイスは、Ｐ＿Ｋｅｙ管理のために、ＩＢ規格で定義されたＳＭＡ属性および方法を利用することができるため、ＳＭは基礎となる実装についての知識がない。

【０１２８】

一実施形態によれば、仮想Ｐ＿Ｋｅｙテーブルに含まれる各Ｐ＿Ｋｅｙは１６ビットを

含むため、そして仮想Ｐ＿Ｋｅｙテーブルに含まれるべき可能性のあるＰ＿Ｋｅｙ値の数（すなわち、最大６４ＫＰ＿Ｋｅｙまで）のため、仮想Ｐ＿Ｋｅｙテーブルのサイズは、スイッチに含まれる、および／またはスイッチによって仮想Ｐ＿Ｋｅｙテーブルに割り当てられるメモリ１４３０の量によって制限され得る。例えば、３６ポートスイッチの各ポートについて６４ＫＰ＿Ｋｅｙを表現するには、必要なメモリサイズは４メガバイトより大きくなり得る。

【０１２９】

図１５は、一実施形態による、パーティション分離のためのビットマップと、レガシー管理エンティティのためのＰ＿Ｋｅｙ抽象化とを有する例示的なファブリックを示す。図に示す実施形態では、スイッチ１４００は、ある数のポート１４２０を備えることができ、これらのポートの各々は、上述したように、Ｐ＿Ｋｅｙ実施のためのビットマップに基づくＰ＿Ｋｅｙテーブル１４１０に関連付けることができる。外部メモリ１５３０は、スイッチ１４００によってアクセス可能である。

【０１３０】

一実施形態によれば、サブネットマネージャ１４４０（例えばレガシーサブネットマネージャ）は、ビットマップに基づくＰ＿Ｋｅｙテーブル１４１０を照会／設定することができないことがあり得る。このような状況において、外部メモリ１５３０内にＰ＿Ｋｅｙテーブル抽象化１４５０（すなわち「仮想Ｐ＿Ｋｅｙテーブル」）を設けることができる。次いで、サブネットマネージャは、サブネットマネージャが連想メモリ／ルックアップに基づいてレガシーＰ＿Ｋｅｙテーブルと対話するであろうのと同じ態様で、仮想Ｐ＿Ｋｅｙテーブルを照会し、それと対話することができる。より具体的には、ＳＭ１４４０は、スイッチ１４００に関連付けられるＳＭＡ１４４１とインターフェイスしてそのような照会を実行することができる。このインターフェイスは、Ｐ＿Ｋｅｙ管理のために、ＩＢ規格で定義されたＳＭＡ属性および方法を利用することができるため、ＳＭは基礎となる実装についての知識がない。

【０１３１】

一実施形態によれば、仮想Ｐ＿Ｋｅｙテーブルに含まれる各Ｐ＿Ｋｅｙは１６ビットを含むため、そして仮想Ｐ＿Ｋｅｙテーブルに含まれるべき可能性のあるＰ＿Ｋｅｙ値の数（すなわち、最大６４ＫＰ＿Ｋｅｙまで）のため、仮想Ｐ＿Ｋｅｙテーブルのサイズは、外部メモリ１５３０の量によって制限され得る。上述した状況とは異なり、外部メモリは、しばしば、スイッチ自体に含まれるメモリの量よりもはるかに大きくなり得る。そのような状況では、仮想Ｐ＿Ｋｅｙテーブルの潜在的なサイズは、図１４に関して上述したものよりもはるかに大きい。そのような場合、たとえば、各スイッチポートに対して６４ＫのＰ＿Ｋｅｙを表現することが可能であろう。

【０１３２】

図１６は、一実施形態による、高性能コンピューティング環境においてスケーラブルなビットマップに基づくＰ＿Ｋｅｙテーブルをサポートするための方法のフローチャートである。ステップ１６１０において、この方法は、１つ以上のマイクロプロセッサを含む１つ以上のコンピュータに少なくとも１つのサブネットを提供することができ、少なくとも１つのサブネットは１つ以上のスイッチを含み、１つ以上のスイッチは少なくともリーフスイッチを含み、１つ以上のスイッチの各々は、複数のスイッチポートを含み、少なくとも１つのサブネットはさらに、複数のホストチャネルアダプタを含み、各ホストチャネルアダプタは、少なくとも１つのホストチャネルアダプタポートを含み、少なくとも１つのサブネットはさらに、複数のエンドノードを含み、複数のエンドノードの各々は、複数のホストチャネルアダプタのうちの少なくとも１つのホストチャネルアダプタに関連付けられる。

【０１３３】

ステップ１６２０において、この方法は、複数の物理ホストおよび仮想マシンの各々を複数のパーティションの少なくとも１つに関連付けることができ、複数のパーティションの各々はＰ＿Ｋｅｙ値に関連付けられる。

【０１３４】
ステップ１６３０において、この方法は、スイッチポートの各々を、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることができる。
【０１３５】
ステップ１６４０において、この方法は、ホストチャネルアダプタポートの各々を、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることができる。
【０１３６】
図１７は、一実施形態による、高性能コンピューティング環境においてビットマップに基づくハードウェア実現例を使用してレガシーＰ＿Ｋｅｙテーブル抽象化をサポートするための方法のフローチャートである。この方法は、１つ以上のマイクロプロセッサを含む１つ以上のコンピュータに少なくとも１つのサブネットを提供することができ、少なくとも１つのサブネットは、１つ以上のスイッチを含み、１つ以上のスイッチは少なくともリーフスイッチを含み、１つ以上のスイッチの各々は、複数のスイッチポートを含み、少なくとも１つのサブネットはさらに、複数のホストチャネルアダプタを含み、各ホストチャネルアダプタは、少なくとも１つのホストチャネルアダプタポートを含み、複数のホストチャネルアダプタは１つ以上のスイッチを介して相互接続され、少なくとも１つのサブネットはさらに、複数のエンドノードを含み、複数のエンドノードの各々は、複数のホストチャネルアダプタのうちの少なくとも１つのホストチャネルアダプタに関連付けられ、少なくとも１つのサブネットはさらに、複数の仮想マシンを含み、複数の仮想マシンの各々は少なくとも１つ仮想機能に関連付けられる。
【０１３７】
ステップ１７２０において、この方法は、複数のエンドノードの各々を複数のパーティションの少なくとも１つに関連付けることができ、複数のパーティションの各々はＰ＿Ｋｅｙ値に関連付けられる。
【０１３８】
ステップ１７３０において、この方法は、１つ以上のスイッチのスイッチポートの各々を、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることができる。
【０１３９】
ステップ１７４０において、この方法は、ホストチャネルアダプタポートの各々を、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることができる。
【０１４０】
ステップ１７５０において、この方法は、複数のビットマップに基づくＰ＿Ｋｅｙテーブルの各々を、仮想Ｐ＿Ｋｅｙテーブルに関連付けることができる。
【０１４１】
一実施形態によれば、高性能コンピューティング環境においてビットマップに基づくハードウェア実現例を使用してレガシーＰ＿Ｋｅｙテーブル抽象化をサポートするためのシステムは、１つ以上のマイクロプロセッサと、少なくとも１つのサブネットとを備え、少なくとも１つのサブネットは、１つ以上のスイッチを含み、１つ以上のスイッチは少なくともリーフスイッチを含み、１つ以上のスイッチの各々は、複数のスイッチポートを含み、少なくとも１つのサブネットはさらに、複数のホストチャネルアダプタを含み、各ホストチャネルアダプタは、少なくとも１つのホストチャネルアダプタポートを含み、少なくとも１つのサブネットはさらに、複数のエンドノードを含み、複数のエンドノードの各々は、複数のホストチャネルアダプタのうちの少なくとも１つのホストチャネルアダプタに関連付けられ、複数のエンドノードの各々は、複数のパーティションのうちの少なくとも１つに関連付けられ、複数のパーティションの各々は、Ｐ＿Ｋｅｙ値に関連付けられ、スイッチポートのうちの１つ以上は、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのう

ちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けられ、ホストチャネルアダプタポートのうちの１つ以上は、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けられ、複数のビットマップに基づくＰ＿Ｋｅｙテーブルの各々は、仮想Ｐ＿Ｋｅｙテーブルに関連付けられる。

【０１４２】

一実施形態によれば、上記のシステムは、さらに、複数のエンドノードの１つを介して動作するサブネットマネージャをさらに備え、サブネットマネージャは、１つ以上のスイッチの各々上の複数のポートを通る許可されたトラフィックおよび許可されないトラフィックを判断する。

【０１４３】

一実施形態によれば、上記のシステムにおいて、サブネットマネージャは、関連付けられる仮想Ｐ＿Ｋｅｙテーブルを介して、１つ以上のスイッチの各々上の複数のポートの各々を通る許可されたトラフィックおよび許可されないトラフィックの判断に基づいて、ビットマップに基づくＰ＿Ｋｅｙテーブルの各々を構成する。

【０１４４】

一実施形態によれば、上記システムにおいて、仮想Ｐ＿Ｋｅｙテーブルの各々は、それぞれのスイッチおよびホストチャネルアダプタにおいてメモリ上にホストされる。

【０１４５】

一実施形態によれば、上記システムにおいて、各仮想Ｐ＿Ｋｅｙテーブルは、それぞれのスイッチおよびホストチャネルアダプタにおけるメモリ量に基づいてサイズが制限される。

【０１４６】

一実施形態によれば、上記のシステムにおいて、各仮想Ｐ＿Ｋｅｙテーブルは、それぞれのスイッチおよびホストチャネルアダプタから外部のメモリ上にホストされる。

【０１４７】

一実施形態によれば、上記のシステムは、２つ以上のサブネットを備え、２つ以上のサブネットの各々は、２つ以上のサブネットの各々における少なくとも１つのルータポートによって相互接続される。

【０１４８】

一実施形態によれば、高性能コンピューティング環境においてビットマップに基づくハードウェア実装を使用してレガシーＰ＿Ｋｅｙテーブル抽象化をサポートするための方法は、１つ以上のマイクロプロセッサを含む１つ以上のコンピュータにおいて、少なくとも１つのサブネットを提供することを備え、少なくとも１つのサブネットは、１つ以上のスイッチを含み、１つ以上のスイッチは少なくともリーフスイッチを含み、１つ以上のスイッチの各々は、複数のスイッチポートを含み、少なくとも１つのサブネットはさらに、複数のホストチャネルアダプタを含み、各ホストチャネルアダプタは、少なくとも１つのホストチャネルアダプタポートを含み、少なくとも１つのサブネットはさらに、複数のエンドノードを含み、複数のエンドノードの各々は、複数のホストチャネルアダプタのうちの少なくとも１つのホストチャネルアダプタに関連付けられ、方法はさらに、複数のエンドノードの各々を複数のパーティションの少なくとも１つに関連付けることを備え、複数のパーティションの各々はＰ＿Ｋｅｙ値に関連付けられ、方法はさらに、１つ以上のスイッチポートの各々を、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることと、ホストチャネルアダプタポートの各々を、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることと、複数のビットマップに基づくＰ＿Ｋｅｙテーブルの各々を、仮想Ｐ＿Ｋｅｙテーブルに関連付けることとを備える。

【０１４９】

一実施形態によれば、上記の方法はさらに、１つ以上のマイクロプロセッサを含む１つ以上のコンピュータにおいて、複数のエンドノードの１つを介して動作するサブネットマネージャを提供することと、サブネットマネージャによって、１つ以上のスイッチの各々

上の複数のポートを通る許可されたトラフィックおよび許可されないトラフィックを判断することとをさらに備える。

【０１５０】

一実施形態によれば、上記の方法はさらに、サブネットマネージャによって、関連付けられる仮想Ｐ＿Ｋｅｙテーブルを介して、１つ以上のスイッチの各々上の複数のポートの各々を通る許可されたトラフィックおよび許可されないトラフィックの判断に基づいて、ビットマップに基づくＰ＿Ｋｅｙテーブルの各々を構成することをさらに備える。

【０１５１】

一実施形態によれば、上記の方法において、仮想Ｐ＿Ｋｅｙテーブルの各々は、それぞれのスイッチおよびホストチャネルアダプタにおいてメモリ上にホストされる。

【０１５２】

一実施形態によれば、上記の方法において、各仮想Ｐ＿Ｋｅｙテーブの各々は、それぞれのスイッチおよびホストチャネルアダプタにおけるメモリ量に基づいてサイズが制限される。

【０１５３】

一実施形態によれば、上記の方法において、仮想Ｐ＿Ｋｅｙテーブルの各々は、それぞれのスイッチおよびホストチャネルアダプタから外部のメモリ上にホストされる。

【０１５４】

一実施形態によれば、上記の方法において、１つ以上のサブネットは、２つ以上のサブネットを備え、２つ以上のサブネットの各々は、２つ以上のサブネットの各々における少なくとも１つのルータによって相互接続される。

【０１５５】

一実施形態によれば、高性能コンピューティング環境においてビットマップに基づくハードウェア実装を使用してレガシーＰ＿Ｋｅｙテーブル抽象化をサポートするための命令をそこに記憶して含む、非一時的なコンピュータ可読記憶媒体であって、命令は、１つ以上のコンピュータによって読み取られ実行されると、１つ以上のコンピュータに、１つ以上のマイクロプロセッサを含む１つ以上のコンピュータに少なくとも１つのサブネットを提供することを含むステップを実行させ、少なくとも１つのサブネットは、１つ以上のスイッチを含み、１つ以上のスイッチは少なくともリーフスイッチを含み、１つ以上のスイッチの各々は、複数のスイッチポートを含み、少なくとも１つのサブネットはさらに、複数のホストチャネルアダプタを含み、各ホストチャネルアダプタは、少なくとも１つのホストチャネルアダプタポートを含み、少なくとも１つのサブネットはさらに、複数のエンドノードを含み、複数のエンドノードの各々は、複数のホストチャネルアダプタのうちの少なくとも１つのホストチャネルアダプタに関連付けられ、命令は、さらに、１つ以上のコンピュータによって読み取られ実行されると、１つ以上のコンピュータに、複数のエンドノードの各々を複数のパーティションの少なくとも１つに関連付けることを含むステップを実行させ、複数のパーティションの各々はＰ＿Ｋｅｙ値に関連付けられ、命令は、さらに、１つ以上のコンピュータによって読み取られ実行されると、１つ以上のコンピュータに、１つ以上のスイッチポートの各々を、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることと、ホストチャネルアダプタポートの各々を、複数のビットマップに基づくＰ＿Ｋｅｙテーブルのうちのあるビットマップに基づくＰ＿Ｋｅｙテーブルに関連付けることと、複数のビットマップに基づくＰ＿Ｋｅｙテーブルの各々を、仮想Ｐ＿Ｋｅｙテーブルに関連付けることとを含むステップを実行させる。

【０１５６】

一実施形態によれば、上記の非一時的なコンピュータ可読記憶媒体において、上記のステップは、さらに、１つ以上のマイクロプロセッサを含む１つ以上のコンピュータにサブネットマネージャを提供することを含み、サブネットマネージャは複数のエンドノードの１つを介して動作し、上記ステップはさらに、サブネットマネージャによって、１つ以上のスイッチの各々上の複数のポートを通る許可されたトラフィックおよび許可されないト

ラフィックを判断することとを含む。
【０１５７】
一実施形態によれば、上記の非一時的なコンピュータ可読記憶媒体において、上記のステップは、さらに、サブネットマネージャによって、関連付けられる仮想Ｐ＿Ｋｅｙテーブルを介して、１つ以上のスイッチの各々上の複数のポートの各々を通る許可されたトラフィックおよび許可されないトラフィックの判断に基づいて、ビットマップに基づくＰ＿Ｋｅｙテーブルの各々を構成することをさらに含む。
【０１５８】
一実施形態によれば、上記の非一時的なコンピュータ可読記憶媒体において、仮想Ｐ＿Ｋｅｙテーブルの各々は、それぞれのスイッチおよびホストチャネルアダプタにおいてメモリ上にホストされる。
【０１５９】
一実施形態によれば、上記の非一時的なコンピュータ可読記憶媒体において、各仮想Ｐ＿Ｋｅｙテーブの各々は、それぞれのスイッチおよびホストチャネルアダプタにおけるメモリ量に基づいてサイズが制限される。
【０１６０】
一実施形態によれば、上記の非一時的なコンピュータ可読記憶媒体において、仮想Ｐ＿Ｋｅｙテーブルの各々は、それぞれのスイッチおよびホストチャネルアダプタから外部のメモリ上にホストされる。
【０１６１】
一実施形態によれば、コンピュータプログラムは、コンピュータシステムによって実行されると、コンピュータシステムに上記の方法を実行させる、機械可読フォーマットにおけるプログラム命令を含む。
【０１６２】
一実施形態によれば、非一時的な機械可読データ記憶媒体に記憶される上記のコンピュータプログラムを備えるコンピュータプログラム製品。
【０１６３】
本発明の多くの特徴は、ハードウェア、ソフトウェア、ファームウェアまたはそれらの組合せにおいて、それらを用いて、またはそれらの支援により、実行可能である。したがって、本発明の特徴は、（たとえば、１つ以上のプロセッサを含む）処理システムを用いて実現され得る。
【０１６４】
この発明の特徴は、ここに提示された特徴のうちのいずれかを行なうように処理システムをプログラミングするために使用可能な命令を格納した記憶媒体またはコンピュータ読取り可能媒体であるコンピュータプログラム製品において、それを使用して、またはその助けを借りて実現され得る。記憶媒体は、フロッピー（登録商標）ディスク、光ディスク、ＤＶＤ、ＣＤ－ＲＯＭ、マイクロドライブ、および光磁気ディスクを含む任意のタイプのディスク、ＲＯＭ、ＲＡＭ、ＥＰＲＯＭ、ＥＥＰＲＯＭ、ＤＲＡＭ、ＶＲＡＭ、フラッシュメモリ装置、磁気カードもしくは光カード、ナノシステム（分子メモリＩＣを含む）、または、命令および／もしくはデータを格納するのに好適な任意のタイプの媒体もしくは装置を含み得るものの、それらに限定されない。
【０１６５】
この発明の特徴は、機械読取り可能媒体のうちのいずれかに格納された状態で、処理システムのハードウェアを制御するために、および処理システムがこの発明の結果を利用する他の機構とやり取りすることを可能にするために、ソフトウェアおよび／またはファームウェアに取込まれ得る。そのようなソフトウェアまたはファームウェアは、アプリケーションコード、装置ドライバ、オペレーティングシステム、および実行環境／コンテナを含み得るものの、それらに限定されない。
【０１６６】
この発明の特徴はまた、たとえば、特定用途向け集積回路（application specific int

egrated circuit：ＡＳＩＣ）などのハードウェアコンポーネントを使用して、ハードウェアにおいて実現されてもよい。ここに説明された機能を行なうようにハードウェアステートマシンを実現することは、関連技術の当業者には明らかであろう。

【０１６７】

加えて、この発明は、この開示の教示に従ってプログラミングされた１つ以上のプロセッサ、メモリおよび／またはコンピュータ読取り可能記憶媒体を含む、１つ以上の従来の汎用または特殊デジタルコンピュータ、コンピューティング装置、マシン、またはマイクロプロセッサを使用して都合よく実現され得る。ソフトウェア技術の当業者には明らかであるように、この開示の教示に基づいて、適切なソフトウェアコーディングが、熟練したプログラマによって容易に準備され得る。

【０１６８】

この発明のさまざまな実施形態が上述されてきたが、それらは限定のためではなく例示のために提示されたことが理解されるべきである。この発明の精神および範囲から逸脱することなく、形状および詳細のさまざまな変更を行なうことができることは、関連技術の当業者には明らかであろう。

【０１６９】

この発明は、特定された機能およびそれらの関係の実行を示す機能的構築ブロックの助けを借りて上述されてきた。説明の便宜上、これらの機能的構築ブロックの境界は、この明細書中ではしばしば任意に規定されてきた。特定された機能およびそれらの関係が適切に実行される限り、代替的な境界を規定することができる。このため、そのようないかなる代替的な境界も、この発明の範囲および精神に含まれる。

【０１７０】

この発明の前述の説明は、例示および説明のために提供されてきた。それは、網羅的であるよう、またはこの発明を開示された形態そのものに限定するよう意図されてはいない。この発明の幅および範囲は、上述の例示的な実施形態のいずれによっても限定されるべきでない。多くの変更および変形が、当業者には明らかになるだろう。これらの変更および変形は、開示された特徴の関連するあらゆる組合せを含む。実施形態は、この発明の原理およびその実用的応用を最良に説明するために選択され説明されたものであり、それにより、考えられる特定の使用に適したさまざまな実施形態についての、およびさまざまな変更例を有するこの発明を、当業者が理解できるようにする。この発明の範囲は、請求項およびそれらの同等例によって定義されるよう意図されている。

【図１】

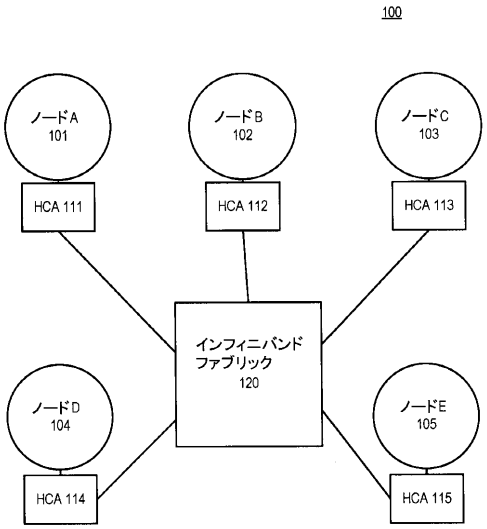

FIGURE 1

【図２】

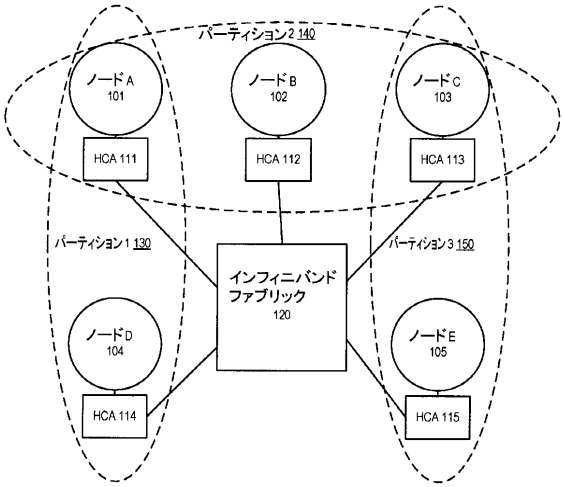

FIGURE 2

【図３】

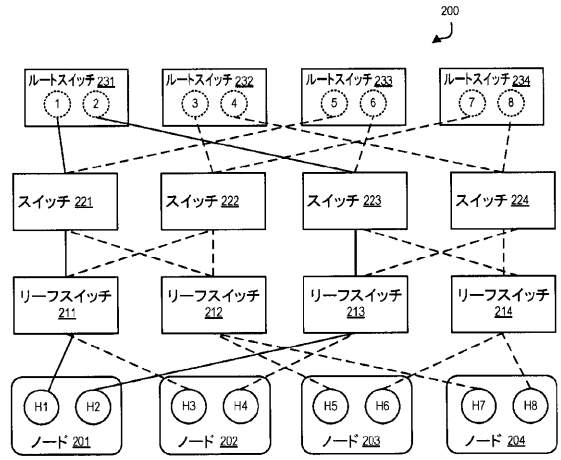

FIGURE 3

【図４】

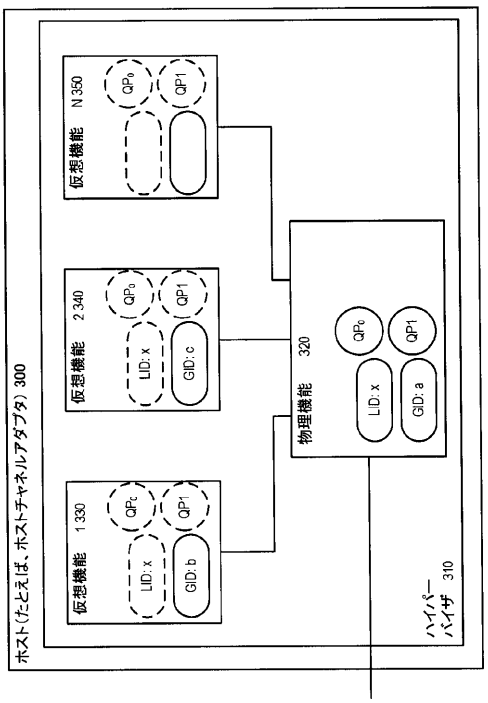

FIGURE 4

【図５】

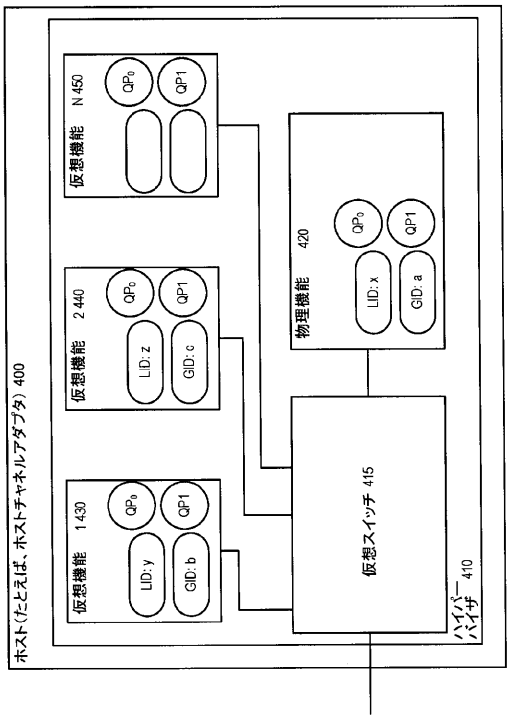

FIGURE 5

【図６】

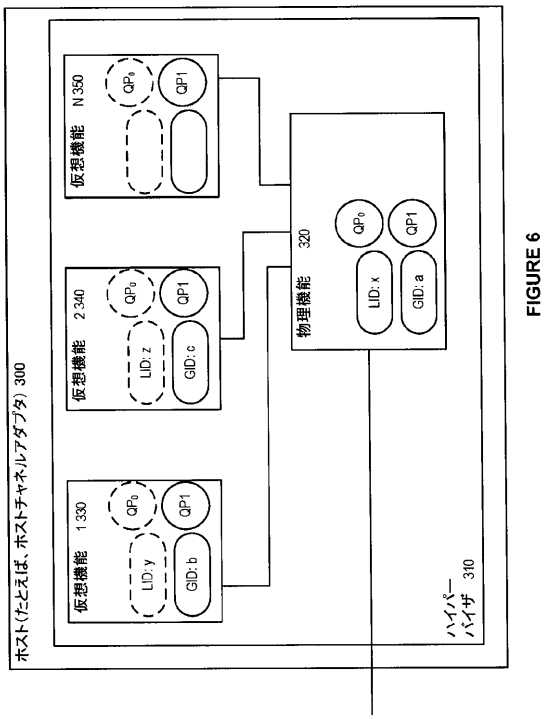

FIGURE 6

【図７】

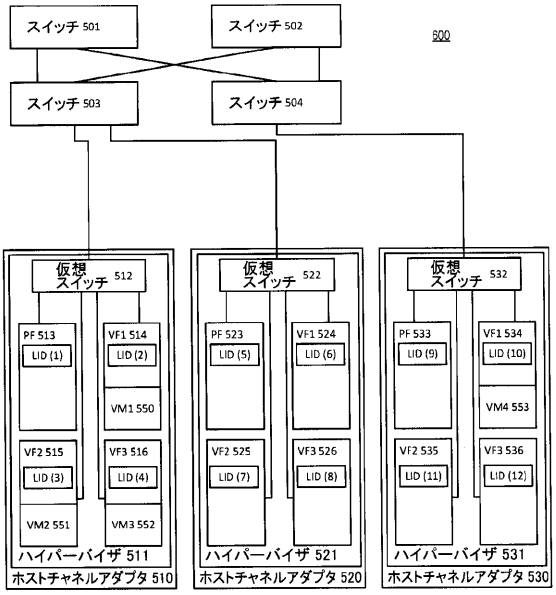

FIGURE 7

【図８】

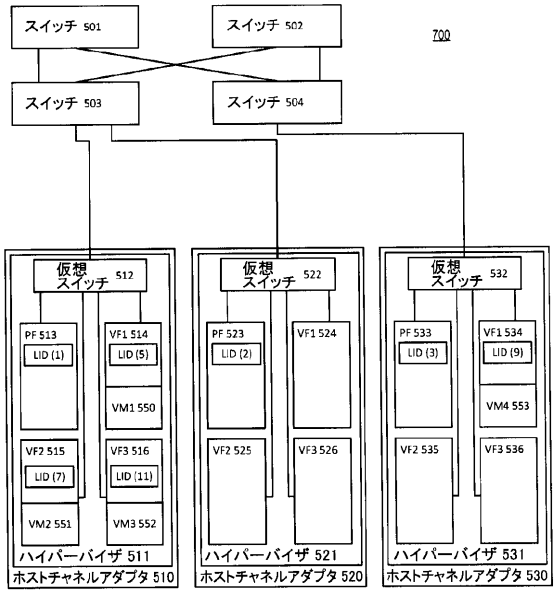

FIGURE 8

【図９】

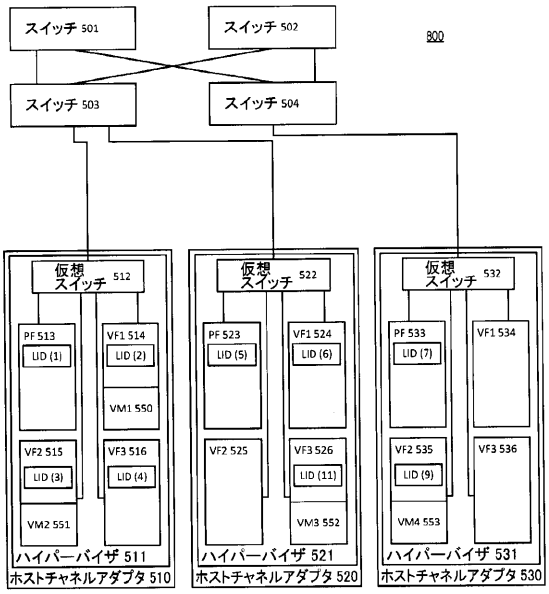

FIGURE 9

【図１０】

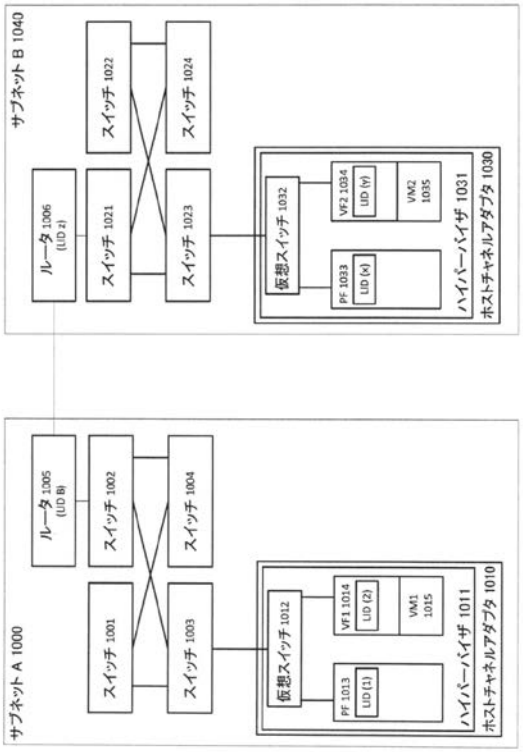

FIGURE 10

【図１１】

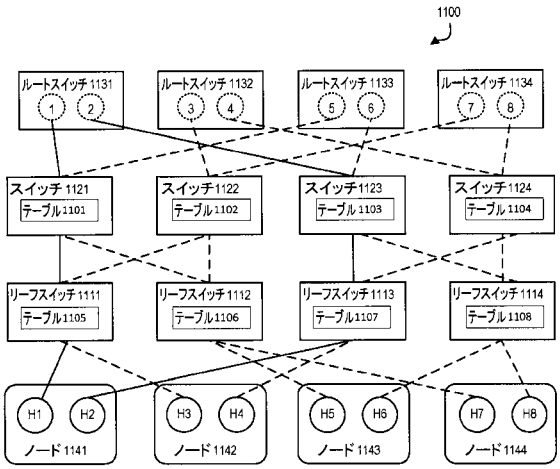

FIGURE 11

【図１２】

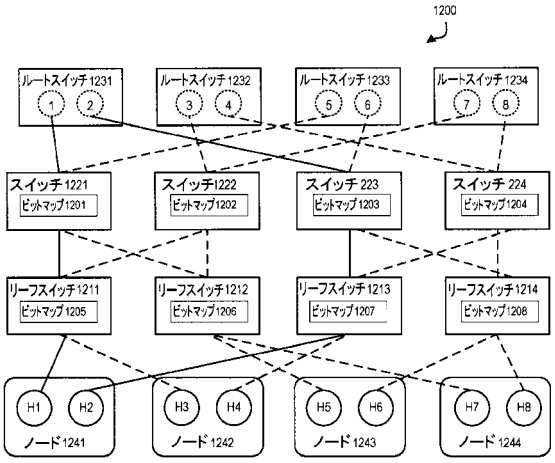

FIGURE 12

【図１３】

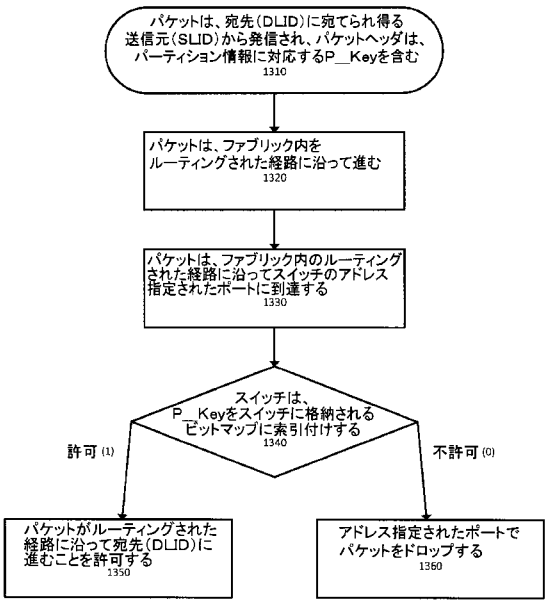

FIGURE 13

【図１４】

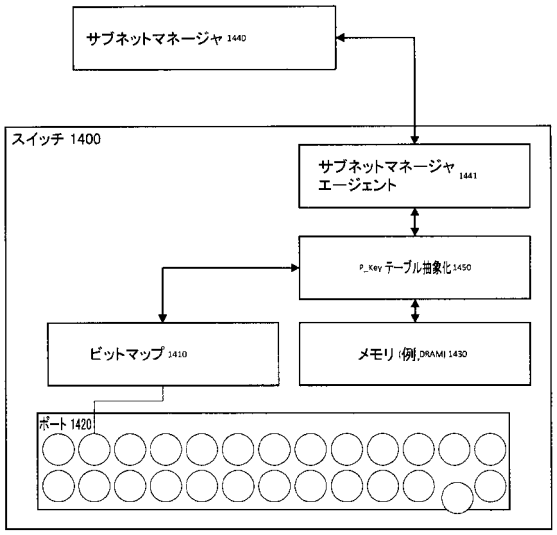

FIGURE 14

【図１５】

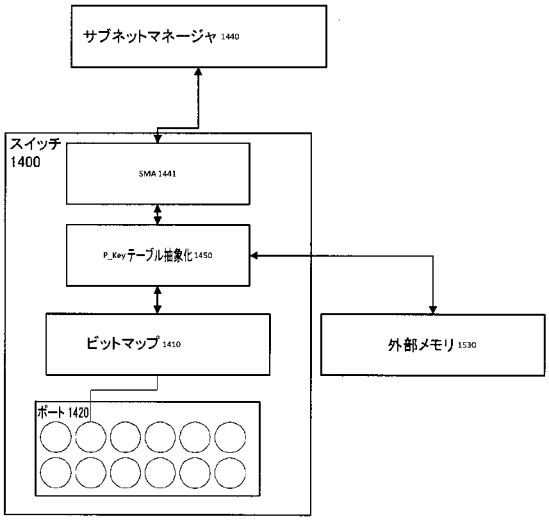

FIGURE 15

【図１６】

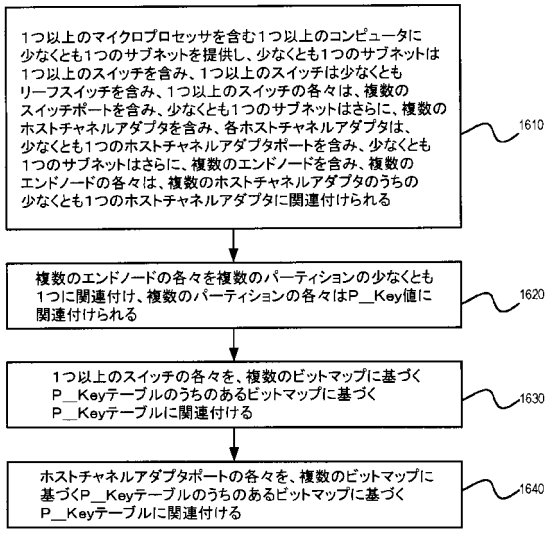

FIGURE 16

【図１７】

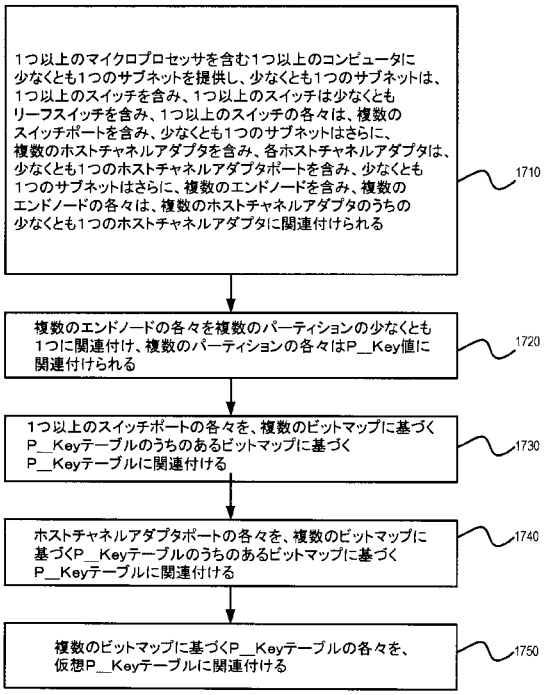

FIGURE 17

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/015156

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L12/931 G06F9/44
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	FEROZ ZAHID ET AL: "Partition-Aware Routing to Improve Network Isolation in Infiniband Based Multi-tenant Clusters", 2015 15TH IEEE/ACM INTERNATIONAL SYMPOSIUM ON CLUSTER, CLOUD AND GRID COMPUTING, 1 May 2015 (2015-05-01), pages 189-198, XP055322241, DOI: 10.1109/CCGrid.2015.96 ISBN: 978-1-4799-8006-2 paragraph [II.B.Infiniband.Architecture.B.Partitioning] ----- -/--	1-22

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
28 March 2017	05/04/2017

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Lefebvre, Laurent

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/015156

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	VISHNU A ET AL: "Performance Modeling of Subnet Management on Fat Tree InfiniBand Networks using OpenSM", PARALLEL AND DISTRIBUTED PROCESSING SYMPOSIUM, 2005. PROCEEDINGS. 19TH IEEE INTERNATIONAL DENVER, CO, USA 04-08 APRIL 2005, PISCATAWAY, NJ, USA,IEEE, 4 April 2005 (2005-04-04), pages 296b-296b, XP010785940, DOI: 10.1109/IPDPS.2005.339 ISBN: 978-0-7695-2312-5 paragraph [2.1.Infiniband.Subnet.Management] paragraph [3.1.Subnet.Discovery] -----	1-22
A	FRANCESCO FUSCO ET AL: "Real-time creation of bitmap indexes on streaming network data", THE VLDB JOURNAL ; THE INTERNATIONAL JOURNAL ON VERY LARGE DATA BASES, SPRINGER, BERLIN, DE, vol. 21, no. 3, 30 July 2011 (2011-07-30), pages 287-307, XP035056143, ISSN: 0949-877X, DOI: 10.1007/S00778-011-0242-X paragraph [4.2.1.Bitmap.Indexes] -----	1,8,15, 21,22

1

Form PCT/ISA/210 (continuation of second sheet) (April 2005)

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,ST,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CZ,DE,DJ,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,JP,KE,KG,KH,KN,KP,KR,KW,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ

(72)発明者 ナラシンハマーシー，プラブフナンダン
アメリカ合衆国、９４０６５ カリフォルニア州、レッドウッド・ショアーズ、オラクル・パークウェイ、５００、エム／エス・５・オゥ・ピィ・７
(72)発明者 ホレン，リネ
ノルウェー、１９００ フェツンド、ビタセン、１７

Ｆターム(参考) 5K030 GA03 HA08 HC13 HD03 KX17 MA14 MD07