(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7109527号
(P7109527)

(45)発行日　令和4年7月29日(2022.7.29)

(24)登録日　令和4年7月21日(2022.7.21)

(51)国際特許分類		FI		
G06F 15/173 (2006.01)		G06F	15/173	685M
G06F 9/455 (2006.01)		G06F	9/455	150
G06F 9/50 (2006.01)		G06F	9/50	120Z
H04L 61/00 (2022.01)		H04L	61/00	
H04L 61/10 (2022.01)		H04L	61/10	

請求項の数　8　外国語出願　（全37頁）

(21)出願番号　特願2020-212507(P2020-212507)
(22)出願日　令和2年12月22日(2020.12.22)
(62)分割の表示　特願2018-501253(P2018-501253)の分割
原出願日　平成28年11月18日(2016.11.18)
(65)公開番号　特開2021-73551(P2021-73551A)
(43)公開日　令和3年5月13日(2021.5.13)
審査請求日　令和3年1月12日(2021.1.12)
(31)優先権主張番号　62/259,321
(32)優先日　平成27年11月24日(2015.11.24)
(33)優先権主張国・地域又は機関　米国(US)
(31)優先権主張番号　62/259,831
(32)優先日　平成27年11月25日(2015.11.25)
(33)優先権主張国・地域又は機関
最終頁に続く

(73)特許権者　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弁理士法人深見特許事務所
(72)発明者　タソウラス，エバンジェロス
ノルウェー、１３２５ リュサケール、ピィ・オゥ・ボックス・１３４
(72)発明者　ザヒド，フェロツ
ノルウェー、１３２５ リュサケール、ピィ・オゥ・ボックス・１３４
(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
最終頁に続く

(54)【発明の名称】　無損失ネットワークにおける効率的な仮想化のためのシステムおよび方法

(57)【特許請求の範囲】

【請求項１】

無損失相互接続ネットワークにおける効率的な仮想化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、

複数のスイッチとを含み、前記複数のスイッチは少なくとも２つのレベルに配置されており、

前記複数のスイッチの各々は、複数のリニアフォワーディングテーブル（linear forwarding table：ＬＦＴ）のうち１つのＬＦＴに関連付けられており、前記システムはさらに、

複数のホストチャネルアダプタを含み、前記複数のホストチャネルアダプタは前記複数のスイッチを介して相互接続されており、前記システムはさらに、

複数のハイパーバイザを含み、前記複数のハイパーバイザの各々は、前記複数のホストチャネルアダプタのうち少なくとも１つのホストチャネルアダプタに関連付けられており、前記システムはさらに、

複数の仮想マシンを含み、前記複数の仮想マシンの各々は、それぞれのハイパーバイザを介して前記複数のホストチャネルアダプタのうち１つのホストチャネルアダプタに接続しており、

前記複数の仮想マシンのうち１つの仮想マシンは、第１のホストチャネルアダプタにおける第１のハイパーバイザから第２のホストチャネルアダプタにおける第２のハイパーバ

イザに対するライブマイグレーションを実行し、前記ライブマイグレーション中に、前記１つの仮想マシンのローカル識別子（local identifier：ＬＩＤ）が更新され、
前記第１のホストチャネルアダプタから前記第２のホストチャネルアダプタに対する前記１つの仮想マシンのマイグレーションの結果、前記複数のＬＦＴのセットが更新され、前記複数のＬＦＴのセットは前記第１のホストチャネルアダプタおよび前記第２のホストチャネルアダプタの最も近い共通の先祖スイッチの判定に基づいて規定されており、
前記仮想マシンの各々には重みパラメータが割当てられており、
前記重みパラメータの各々は前記複数のＬＦＴを計算するのに用いられ、
前記複数のＬＦＴを計算することは、
前記仮想マシンのために、利用可能な上りポート群の間で最少累積の下り重みに基づいてスイッチにおける下りポートを選択することと、
前記仮想マシンの割当てられた前記重みパラメータによって、前記選択されたポートについて、累積された下り重みを増やすこととを含む、システム。
【請求項２】
無損失相互接続ネットワークにおける効率的な仮想化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おいて、複数のスイッチと、複数のホストチャネルアダプタと、複数のハイパーバイザと、複数の仮想マシンとを設けるステップを含み、
前記複数のスイッチは少なくとも２つのレベルに配置されており、前記複数のスイッチの各々は、複数のリニアフォワーディングテーブル（ＬＦＴ）のうち１つのＬＦＴに関連付けられており、
前記複数のホストチャネルアダプタは前記複数のスイッチを介して相互接続されており、
前記複数のハイパーバイザの各々は、前記複数のホストチャネルアダプタのうち少なくとも１つのホストチャネルアダプタに関連付けられており、
前記複数の仮想マシンの各々は、それぞれのハイパーバイザを介して前記複数のホストチャネルアダプタのうち１つのホストチャネルアダプタに接続しており、前記方法はさらに、
第１のホストチャネルアダプタにおける第１のハイパーバイザから第２のホストチャネルアダプタにおける第２のハイパーバイザに対する前記複数の仮想マシンのうち１つの仮想マシンのライブマイグレーションを実行するステップを含み、前記ライブマイグレーション中に、前記１つの仮想マシンのローカル識別子（ＬＩＤ）が更新され、前記方法はさらに、
前記第１のホストチャネルアダプタから前記第２のホストチャネルアダプタに対する前記１つの仮想マシンの前記ライブマイグレーションの結果、前記複数のＬＦＴのセットを更新するステップを含み、前記複数のＬＦＴのセットは前記第１のホストチャネルアダプタおよび前記第２のホストチャネルアダプタの最も近い共通の先祖スイッチの判定に基づいて規定されており、
前記仮想マシンの各々には重みパラメータが割当てられており、
前記重みパラメータの各々は前記複数のＬＦＴを計算するのに用いられ、
前記複数のＬＦＴを計算することは、
前記仮想マシンのために、利用可能な上りポート群の間で最少累積の下り重みに基づいてスイッチにおける下りポートを選択することと、
前記仮想マシンの割当てられた前記重みパラメータによって、前記選択されたポートについて、累積された下り重みを増やすこととを含む、方法。
【請求項３】
前記第１のホストチャネルアダプタおよび前記第２のホストチャネルアダプタの前記最も近い共通の先祖スイッチの前記判定は、
前記第１のホストチャネルアダプタから前記第２のホストチャネルアダプタに向かって上方向に第１の経路をトレースするステップと、

前記第２のホストチャネルアダプタから前記第１のホストチャネルアダプタに向かって上方向に第２の経路を同時にトレースするステップと、
印付けメカニズムによって、前記第１の経路と前記第２の経路とが合流する地点において前記最も近い共通の先祖スイッチに印付けするステップとを含む、請求項２に記載の方法。
【請求項４】
前記第１の経路および前記第２の経路に沿った各々のスイッチは前記印付けメカニズムによって印付けされる、請求項３に記載の方法。
【請求項５】
前記複数のＬＦＴのセットは、印付けされた前記最も近い共通の先祖スイッチにおける各々のＬＦＴと、前記第１の経路および前記第２の経路に沿って印付けされた各々のスイッチの各々のＬＦＴとを含む、請求項４に記載の方法。
【請求項６】
前記重みパラメータの各々は、仮想スイッチとリーフスイッチとの間の各リンクのトラフィックの割合を反映している、請求項２～５のいずれか１項に記載の方法。
【請求項７】
前記複数のＬＦＴの各々は宛先ノードから計算が開始される、請求項６に記載の方法。
【請求項８】
無損失相互接続ネットワークにおける効率的な仮想化をサポートするための命令が格納されているコンピュータ読取り可能プログラムであって、前記命令が１つ以上のコンピュータによって読出されて実行されると、前記１つ以上のコンピュータに請求項２～７のいずれか１項に記載の方法を実行させる、コンピュータ読取り可能プログラム。
【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。この特許文献または特許開示は特許商標庁の特許ファイルまたは記録に記載されているため、著作権保有者は、何人によるその複写複製に対しても異議はないが、その他の場合には如何なるときもすべての著作権を保有する。
【０００２】
発明の分野：
本発明は、概して、コンピュータシステムに関し、特に、ＳＲ－ＩＯＶ ｖＳｗｉｔｃｈアーキテクチャを用いてコンピュータシステム仮想化およびライブマイグレーション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：
サブネットにおいて仮想マシンマイグレーションをサポートするためのシステムおよび方法がこの明細書中に記載される。例示的な方法は、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、１つ以上のスイッチを設けることができ、当該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、当該１つ以上のスイッチの各々は複数のポートを含み、当該方法はさらに、複数のホストチャネルアダプタを設けることができる。複数のホストチャネルアダプタの各々は、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。複数のホストチャネルアダプタは当該１つ以上のスイッチを介して相互接続されている。当該方法はさらに、複数のハイパーバイザを設けることができる。当該複数のハイパーバイザの各々は当該複数のホストチャネルアダプタのうち少なくとも１つのホストチャネルアダプタに関連付けられている。当該方法はさらに、複数の仮想マシンを設けることができる。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。当該方法はさらに、予めポピュレートされたローカル識別子（local identifier：ＬＩＤ）アーキテクチャを備えた仮想スイッチ、または動的ＬＩＤ割当てアーキテクチャを備えた仮想スイッチのうち１つ以上を備えた複数のホストチャネルアダプタを配置することができる。当該方法は、各々の仮想スイッチをＬＩＤに割当てることができ、割当てられたＬＩＤは関連付けられた物理機能のＬＩＤに対応している。当該方法は、仮想スイッチの各々に割当てられたＬＩＤに少なくとも基づいて、１つ以上のリニアフォワーディングテーブル（linear forwarding table：ＬＦＴ）を計算することができる。１つ以上のＬＦＴの各々は、１つ以上のスイッチのうちの一のスイッチに関連付けられている。

【０００５】

一実施形態に従うと、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、１つ以上のマイクロプロセッサと、少なくともリーフスイッチを含む１つ以上のスイッチとを設けることができ、当該１つ以上のスイッチの各々は複数のポートを含み、さらに、複数のホストチャネルアダプタを設けることができ、ホストチャネルアダプタの各々は少なくとも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は１つ以上のスイッチを介して相互接続されており、さらに、複数のハイパーバイザを設けることができ、複数の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ており、さらに、複数の仮想マシンを設けることができ、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。当該方法は、予めポピュレートされたローカル識別子（ＬＩＤ）アーキテクチャを備えた仮想スイッチまたは動的ＬＩＤ割当てアーキテクチャを備えた仮想スイッチのうち１つ以上を備えた複数のホストチャネルアダプタを配置することができる。当該方法は、仮想スイッチの各々に複数の物理的ＬＩＤ（physical LID：ｐＬＩＤ）のうち１つのｐＬＩＤを割当てることができ、割当てられたｐＬＩＤは関連付けられた物理機能のｐＬＩＤに対応している。当該方法はまた、複数の仮想マシンの各々に複数の仮想ＬＩＤ（virtual LID：ｖＬＩＤ）のうち１つのｖＬＩＤを割当てることができ、ＬＩＤスペースは複数のｐＬＩＤおよび複数のｖＬＩＤを含んでいる。

【０００６】

一実施形態に従うと、各々のｐＬＩＤ値は、インフィニバンドパケットのローカルルートヘッダにおける標準ＳＬＩＤフィールドおよび標準ＤＬＩＤフィールドを用いて表わすことができる。同様に、各々のｖＬＩＤ値は、拡張を表わす追加の２ビット以上と組合わせて、標準ＳＬＩＤフィールドと標準ＤＬＩＤフィールドとの組合せを用いて表わすことができる。

【図面の簡単な説明】

【０００７】

【図１】一実施形態に従ったインフィニバンド環境の一例を示す図である。

【図２】一実施形態に従った、ネットワーク環境におけるツリートポロジーの一例を示す図である。

【図３】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４】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５】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６】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７】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８】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９】一実施形態に従った、拡張されたローカルルートヘッダを示す図である。
【図１０】一実施形態に従った、２つの例示的なリニアフォワーディングテーブルを示す図である。
【図１１】一実施形態に従った、無損失相互接続ネットワークにおける効率的な仮想化のサポートの例を示す図である。
【図１２】一実施形態に従った、無損失相互接続ネットワークにおける効率的な仮想化のサポートの例を示す図である。
【図１３】一実施形態に従った、無損失相互接続ネットワークにおける効率的な仮想化のサポートの例を示す図である。
【図１４】一実施形態に従った、無損失相互接続ネットワークにおける効率的な仮想化のサポートの例を示す図である。
【図１５】一実施形態に従った潜在的な仮想マシンマイグレーションを示す図である。
【図１６】一実施形態に従ったスイッチタプルを示す図である。
【図１７】一実施形態に従った再構成プロセスを示す図である。
【図１８】一実施形態に従った、無損失相互接続ネットワークにおける効率的な仮想化をサポートするための方法を示すフローチャートである。
【図１９】一実施形態に従った、無損失相互接続ネットワークにおける効率的な仮想化をサポートするための方法を示す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】
無損失相互接続ネットワークにおける効率的な仮想化をサポートするためのシステムおよび方法がこの明細書中に記載される。
【００１１】
この発明の以下の説明は、高性能ネットワークについての一例として、インフィニバンド（ＩＢ）ネットワークを使用する。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。

【００２５】

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。

【００２６】

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。

【００２７】

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。

【００２８】

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。

【００２９】

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。

【００３０】

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。

【００３１】

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。

【００３２】

図２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２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。

【００３３】

また、図２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。

【００３４】

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。

【００３５】

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。

【００３６】

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。

【００３７】

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。

【００３８】

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。

【００３９】

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。

【００４０】

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。

【００４１】

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。

【００４２】

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。

【００４３】

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。

【００４４】

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。

【００４５】

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。

【００４６】

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。

【００４７】

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。

【００４８】

<u>インフィニバンドＳＲ－ＩＯＶアーキテクチャ－共有ポート</u>

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。

【００４９】

図３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。

【００５０】

一実施形態に従うと、図３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。

【００５１】

図３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。

【００５２】

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。

【００５３】

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。

【００５４】

<u>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）</u>

図４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。

【００５５】

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 Adapter：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。

【００５６】

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。

【００５７】

<u>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）</u>

図５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。

【００５８】

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。

【００５９】

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。

【００６０】

図６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。

【００６１】

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。

【００６２】

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。

【００６３】

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５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５１４～５

１６、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５に示されるようにＬＩＤが割当てられている。

【００６４】

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。

【００６５】

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。

【００６６】

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。

【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。

【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。

【００６９】

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。

【００７０】

図７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。

【００７１】

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。

【００７２】

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。

【００７３】

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７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。Ｌ

ＩＤ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。

【００７４】

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。

【００７５】

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。

【００７６】

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。

【００７７】

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。

【００７８】

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。

【００７９】

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。

【００８０】

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８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。

【００８１】

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。

【００８２】

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。

【００８３】

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７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。

【００８４】

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８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。

【００８５】

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。

【００８６】

ｖＳｗｉｔｃｈスケーラビリティ

一実施形態に従うと、ｖＳｗｉｔｃｈアーキテクチャを用いる場合の問題はＬＩＤスペースが制限されていることである。ＬＩＤスペースに関するスケーラビリティの問題を克服するために、以下の３つの代替例（各々を以下にさらに詳細に説明する）を独立して用いるかまたは組合わせることができる：すなわち、複数のサブネットを用いること；後方互換性のあるＬＩＤスペース拡張を導入すること；および、軽量のｖＳｗｉｔｃｈを形成するためにｖＰｏｒｔアーキテクチャとｖＳｗｉｔｃｈアーキテクチャとを組合わせること；である。

【００８７】

一実施形態に従うと、複数のＩＢサブネットを用いることができる。ＬＩＤは、層２アドレスであり、サブネット内において固有でなければならない。ＩＢトポロジーが複数のサブネット上にわたっている場合、ＬＩＤはそれ以上制限事項とはならないが、ＶＭを異なるサブネットにマイグレートする必要がある場合、そのＬＩＤアドレスは変更することができる。なぜなら、そのアドレスが新しいサブネットにおいて既に使用されているかもしれないからである。複数のサブネット上にわたっていることで、単一のサブネットトポロジーのＬＩＤ制限を解決することができるが、これはまた、ルーティングプロセスに付加的なオーバーヘッドおよび待ち時間を付加するサブネット間ルーティングのために層３ＧＩＤアドレスを用いなければならないことを意味している。なぜなら、サブネットの端に位置するルータによって層２ヘッダを変更しなければならないからである。また、現在のハードウェア実装、ソフトウェア実装および緩いＩＢＡ（インフィニバンド・アーキテクチャ）規格の下では、複数のサブネット上にわたっているクラスタのために最適化されたルーティング経路を提供するために、個々のサブネットのＳＭはグローバルトポロジーを認識することができなくなっている。

【００８８】

一実施形態に従うと、ＩＢＡにおける後方互換性のあるＬＩＤスペース拡張を導入することができる。ＬＩＤビットの数を、たとえば２４ビットまたは３２ビットに増やすことにより、不十分なＬＩＤスペースを増やす場合に問題が生じる可能性がある。このような量だけＬＩＤスペースを増やすことにより、後方互換性に破断が生じる可能性がある。な

ぜなら、ＩＢローカルルートヘッダ（Local Route Header：ＬＲＨ）がオーバーホールされなければならなくなり、レガシーハードウェアが新しい基準では機能することができなくなるからである。一実施形態に従うと、後方互換性を維持しながらも、依然として新しいハードウェアが拡張機能を利用できるように、ＬＩＤスペースを拡張することができる。ＬＲＨは、０として送信されて受信機には無視される予備の７ビットを有する。送信元ＬＩＤ（Source LID：ＳＬＩＤ）についてのＬＲＨにおけるこれらの予備ビットと宛先ＬＩＤ（Destination LID：ＤＬＩＤ）についての２ビットとのうち２つを利用することにより、ＬＩＤスペースを１８ビットに拡張する（ＬＩＤスペースを４倍にする）ことができ、物理的装置に割当てられた物理的ＬＩＤ（ｐＬＩＤ）およびＶＭに割当てられた仮想ＬＩＤ（ｖＬＩＤ）を用いたスキームを作成することができる。

【００８９】

一実施形態に従うと、追加の２ビットが０として送信されると、ＬＩＤがＩＢＡ（４８ＫユニキャストＬＩＤおよび１６Ｋマルチキャスト ＬＩＤ）においてその時点で定義されるとおりに用いられ、スイッチは、パケットの転送のためにそれらの主要なＬＦＴを検索することができる。他の場合、ＬＩＤはｖＬＩＤであり、１９２Ｋのサイズを有する二次的ＬＦＴに基づいて転送することができる。ｖＬＩＤがＶＭに属しており、ＶＭが、ｐＬＩＤを有する物理ノードとアップリンクを共有しているので、ｖＬＩＤは、ネットワークを構成（たとえば初期構成）または（たとえばトポロジー変更後に）再構成する際に、経路演算段階から除外することができるが、スイッチにおける二次的ＬＦＴテーブルは、上述のとおり更新することができる。ＳＭがネットワークをブートし発見すると、ＳＭはハードウェアのすべてが拡張されたＬＩＤスペースをサポートするかどうかを識別することができる。そうでなければ、ＳＭはレガシー互換モードでフォールバックすることができ、ＶＭはｐＬＩＤスペースからのＬＩＤを占有するはずである。

【００９０】

図９は、一実施形態に従った、拡張されたローカルルートヘッダを示す。図に示されるように、ローカルルートヘッダ内では、仮想レーン（virtual lane：ＶＬ）９００は４ビットを含み、リンクバージョン（link version：Ｌｖｅｒ）９０１は４ビットを含み、サービスレベル（service level：ＳＬ）９０２は４ビットを含み、ＬＩＤ拡張フラグ（LID extension flag：ＬＥＸＴＦ）９０３は１ビットを含み、第１の予備ビット（Ｒ１）９０４は１ビットを含み、リンク次ヘッダ（link next header：ＬＮＨ）９０５は２ビットを含み、宛先ローカルＩＤ（destination local ID：ＤＬＩＤ）９０６は１６ビットを含み、ＤＬＩＤプレフィックス拡張（DLID prefix extension：ＤＰＦ）９０７は２ビットを含み、ＳＬＩＤプレフィックス拡張（SLID prefix extension：ＳＰＦ）９０８は２ビットを含み、第２の予備ビット（Ｒ２）９０９は１ビットを含み、パケット長（packet length：ＰｋｔＬｅｎ）９１０は１１ビット含み、送信元ローカルＩＤ（source local ID：ＳＬＩＤ）９１１は１６ビットを含む。一実施形態に従うと、両方の予備ビット９０４および９０９はゼロに設定することができる。

【００９１】

一実施形態に従うと、上述のとおり、図９に示されるＬＲＨは、宛先ローカルＩＤ９０６および送信元ローカルＩＤ９０８についてのプレフィックス拡張として７つの（元の）予備ビットのうちの４つを利用する。これにより、利用時に、ＬＩＤ拡張フラグに関連付けて、スイッチにおける二次的ＬＦＴを介してルーティングされ得るｖＬＩＤに関連付けてＬＲＨが使用されることが信号で伝えられる。代替的には、拡張９０７および９０８がゼロとして送信され（受信機によって無視され）ると、ＬＩＤは、ｐＬＩＤに関連付けられ、ＩＢＡにおいてその時点で定義されるとおり用いられる。

【００９２】

図１０は、一実施形態に従った、２つの例示的なリニアフォワーディングテーブルを示す。図１０に示されるように、リニアフォワーディングテーブル９１６はｐＬＩＤに関連付けられたフォワーディングテーブルである。ＬＦＴは、エントリ９１２（ＤＬＩＤ＝０によって索引付けされたエントリ０）からエントリ９１３（ＤＬＩＤ＝４８Ｋ－１によっ

て索引付けされたエントリ４８Ｋ－１）にわたっている。この場合、ＬＦＴにおける各エントリは、規格１６ビットＤＬＩＤによって索引付けされ、標準ＩＢポート番号を含んでいる。対照的に、リニアフォワーディングテーブル９１７はｖＬＩＤに関連付けられた二次的フォワーディングテーブルである。ＬＦＴは、エントリ９１４（１８ビットＤＰＦ＋ＤＬＩＤ＝０によって索引付けされたエントリ０）からエントリ９１５（１８ビットＤＰＦ＋ＤＬＩＤ＝２５６Ｋ－１によって索引付けされたエントリ２５６Ｋ－１）にわたっている。この場合、各エントリは、拡張された１８ビットＤＰＦ＋ＤＬＩＤによって索引付けされ、標準ＩＢポート番号を含んでいる。

【００９３】

一実施形態に従うと、軽量のｖＳｗｉｔｃｈアーキテクチャを形成するためにハイブリッドアーキテクチャを用いることができる。マイグレートされたＶＭと共にＬＩＤをマイグレートすることができるｖＳｗｉｔｃｈアーキテクチャは、ＬＩＤが変化するであろう共有のＬＩＤのスキームとは対照的にマイグレーションの後にピアとの接続性を再構築するために付加的なシグナリングについての要件が存在しないので、サブネット管理に対して十分にスケーリングする。他方で、共有のＬＩＤスキームは、ＬＩＤスペースに対して十分にスケーリングする。ハイブリッドｖＳｗｉｔｃｈ＋共有型ｖＰｏｒｔモデルは、ＳＭがサブネットにおける利用可能なＳＲ－ＩＯＶ仮想機能を認識する場合、実現することができるが、特定のＶＦが専用のＬＩＤを受取り得る一方で、他のものはそれらのＧＩＤに基づいて共有ＬＩＤの態様でルーティングされている。ＶＭノード役割についての何らかの情報があれば、（たとえば、ルートを計算し、ネットワークにおける負荷バランシングを実行している間に別々に考慮されるようにするために）、多数のピアを備えたポピュラーなＶＭ（たとえばサーバ）には専用のＬＩＤが割当てられ得る一方で、多くのピアと対話しないかまたはステートレスなサービスを実行する（マイグレートされる必要がなく、再生成され得る）他のＶＭはＬＩＤを共有することができる。

【００９４】

ｖＳｗｉｔｃｈベースのサブネットのためのルーティング戦略

一実施形態に従うと、より高い性能を得るために、ルーティングアルゴリズムは、ルートを計算する際にｖＳｗｉｔｃｈアーキテクチャを考慮に入れることができる。ファットツリーにおいては、ｖＳｗｉｔｃｈは、ｖＳｗｉｔｃｈが対応するリーフスイッチへの上りリンクを１つだけ有するという独特な特性によってトポロジー発見プロセスにおいて識別することができる。ｖＳｗｉｔｃｈが識別されると、ルーティング機能は、各ＶＭからのトラフィックがネットワークにおける他のすべてのＶＭに向かう経路を発見することができるように、すべてのスイッチのためのＬＦＴを生成することができる。各ＶＭはそれ自体のアドレスを有しており、このため、各ＶＭは、同じｖＳｗｉｔｃｈに付与された他のＶＭからは独立してルーティングすることができる。これにより、結果として、トポロジーにおけるｖＳｗｉｔｃｈに向かうとともに各々が特定のＶＭへのトラフィックを担持している独立した複数の経路を生成するルーティング機能が得られる。このアプローチの１つの欠点として、ＶＭ分配がｖＳｗｉｔｃｈの間で均一でない場合、より多くのＶＭを備えたｖＳｗｉｔｃｈには潜在的により大きなネットワークリソースが割当てられる点がある。しかしながら、ｖＳｗｉｔｃｈから対応するリーフスイッチまでの単一の上りリンクは、依然として、特定のｖＳｗｉｔｃｈに付与されたすべてのＶＭによって共有されるボトルネックリンクのままである。結果として、準最適にネットワークが利用される可能性がある。最も単純で最速のルーティング戦略は、すべてのｖＳｗｉｔｃｈ－ｖＳｗｉｔｃｈの対の間に経路を生成して、対応するｖＳｗｉｔｃｈに割当てられるのと同じ経路を備えたＶＭをルーティングすることである。予めポピュレートされたＬＩＤ割当てスキームと動的ＬＩＤ割当てスキームとがあれば、各々のｖＳｗｉｔｃｈは、ＳＲ－ＩＯＶアーキテクチャにおけるＰＦによって定義されたＬＩＤを有する。ｖＳｗｉｔｃｈについてのこれらのＰＦ　ＬＩＤは、ルーティングの第１段階でＬＦＴを生成するために用いることができ、第２段階では、ＶＭのＬＩＤを生成されたＬＦＴに追加することができる。予めポピュレートされたＬＩＤスキームにおいては、ＶＦ　ＬＩＤへのエントリは対応するｖ

Ｓｗｉｔｃｈの出力ポートをコピーすることによって追加することができる。同様に、新しいＶＭがブートされた場合の動的ＬＩＤ割当ての場合、ＶＭのＬＩＤと対応するｖＳｗｉｔｃｈによって決定された出力ポートとを備えた新しいエントリがすべてのＬＦＴにおいて追加される。この戦略についての問題点は、ｖＳｗｉｔｃｈを共有する別々のテナントに属するＶＭが、ネットワークにおいて同じ完全な経路を共有しているせいで、それらの間で固有に干渉する可能性がある点である。高いネットワーク利用率を維持しながらもこの問題を解決するために、仮想化されたサブネットのための重み付けされたルーティングスキームを用いることができる。

【００９５】

一実施形態に従うと、ｖＳｗｉｔｃｈベースの仮想化サブネットのための重み付けされたルーティングスキームを利用することができる。このようなメカニズムにおいては、ｖＳｗｉｔｃｈ上の各ＶＭには、ルートを計算する際にバランスを取るために考慮に入れることができるパラメータ重みが割当てられる。重みパラメータの値は、そのｖＳｗｉｔｃｈにおけるＶＭに割付けられたリーフスイッチリンク容量に対するｖＳｗｉｔｃｈの割合を反映している。たとえば、単純な構成により、各ＶＭに、１／ｎｕｍ＿ｖｍｓに等しい重みが割当てられてもよく、この場合、ｎｕｍ＿ｖｍｓは、対応するｖＳｗｉｔｃｈハイパーバイザ上のブートされたＶＭの数である。別の可能な実現例は、最も重要なＶＭに対して、これらＶＭに向かって流れるトラフィックに優先順位を付けるために、より高い割合のｖＳｗｉｔｃｈ容量を割当てることであり得る。しかしながら、ｖＳｗｉｔｃｈ毎のＶＭの累積的な重みはすべてのｖＳｗｉｔｃｈ上で等しくなり得るので、トポロジーにおけるリンクは、実際のＶＭ分配によって影響されことなくバランスを取ることができる。同時に、スキームは、トポロジーにおける中間リンクで同じｖＳｗｉｔｃｈ　ＶＭ間における干渉をなくした上で、各ＶＭがネットワークにおいて独立してルーティングされ得る多重通路を可能にする。当該スキームは、ＶＭがその割当てられた容量を上回るのを確実に防止するために、ＶＭ率の上限ごとに、各ｖＳｗｉｔｃｈ上での実施と組合わせることができる。加えて、ネットワークにおいて複数のテナントグループが存在している場合、テナント認識型ルーティングのような技術は、テナント間でネットワーク全体を分離させるために、提案されたルーティングスキームと統合することができる。

【００９６】

一実施形態に従うと、以下に、ＩＢベースのファットツリートポロジーについての重み付けされたルーティングを記載する。ファットツリールーティングアルゴリズムとして、ｖＳｗｉｔｃｈＦａｔＴｒｅｅは、サブネットにおける各ＶＭに関連付けられたＬＩＤのためのすべてのスイッチにおけるＬＦＴを設定するために、ファットツリートポロジーを再帰的に横断する。このメカニズムは決定論的であり、すべてのルートについての後方計算が宛先ノードから開始される宛先ベースのルーティングをサポートする。

【００９７】

仮想化されたサブネットについての重み付けされたファットツリールーティングアルゴリズム

【００９８】

【数１】

```
1: procedure ROUTEVIRTUALIZEDNODES
2:     for all s ∈ leafSwitches[] do
3:         sort vswitches in the increasing order of connected virtual
           machines
4:         for all v ∈ vSwitches[] do
5:           num_vms  ← GETTOTALVMS(v)
6:           vm_weight ← 1/num_vms
7:           for all vm ∈ vSwitches[] do
8:             vm.weight ←  vm_weight
9:             s.LFT[vm.LID] ← v.port
10:            ROUTEDOWNGOINGBYGOINGUP(s,vm)
11:          end for
12:        end for
13:    end for
14: end procedure
15: procedure ROUTEDOWNGOINGBYGOINGUP(s,vm)
16:    p ← GETLEASTLOADEDPORT(s.UpGroups[])
17:    rSwitch ←  p.Switch
18:    rSwitch.LFT[vm.LID] ← p
19:    p.Dwn += vm.weight
20:    ROUTEUPGOINGBYGOINGDOWN(s,vm)
21:    ROUTEDOWNGOINGBYGOINGUP(rSwitch,vm)
22: end procedure
23: procedure ROUTEUPGOINGBYGOINGDOWN(s,vm)
24:    for all g ∈ s.DownGroups[] do
25:       skip g if the LFT(vm.LID) is part of this group
26:       p  ← GETLEASTLOADEDPORT(g)
27:       rSwitch  ←  p.Switch
28:       rSwitch.LFT[vm.LID] ←  p
29:       p.Up += vm.weight
30:       ROUTEUPGOINGBYGOINGDOWN(rSwitch, vm)
31:    end for
32: end procedure
```

【００９９】

一実施形態に従うと、ｖＳｗｉｔｃｈＦａｔＴｒｅｅルーティングメカニズムは以下のように作用する。各々のＶＭには、比例した重みが割当てられる。この比例した重みは、ｖＳｗｉｔｃｈノードの（たとえば、定数１として得られる）重みをその上で実行されるＶＭの総数で割ることによって計算される。さまざまな重み付けスキームを実現することもできる。たとえば、ＶＭタイプに基づいて重みを割当てるための実現例を選ぶことができる。しかしながら、簡潔にするために、この説明は比例重み付けスキームに焦点を合わせている。各々のリーフスイッチのために、ルーティングメカニズムは、接続されたＶＭ（行３）に基づいて減少する順序で、接続されたｖＳｗｉｔｃｈをソートする。この順序

は、より高い重みが付けられたＶＭが最初にルーティングされることを確実にするので、リンクに割当てられたルートのバランスを取ることができる。ルーティングメカニズムは、すべてのリーフスイッチおよびそれらの対応するｖＳｗｉｔｃｈを通過し、各々のＶＭからツリー内を横断して、ＲＯＵＴＥＤＯＷＮＧＯＩＮＧＢＹＧＯＩＮＧＵＰ（行１０）をコールすることによって、ツリー内においてＶＭに向かう経路を再帰的に割当てる。各々のスイッチにおける下りポートは、利用可能な上りポート群のすべての中で最少累積の下り重み（downward weight）に基づいて選択されている（ＲＯＵＴＥＤＯＷＮＧＯＩＮＧＢＹＧＯＩＮＧＵＰ；行１６）。下りポートが選択されると、当該メカニズムは、ルーティングされているＶＭの重みによって、対応するポートについての下り累積重みを増やすことができる（ＲＯＵＴＥＤＯＷＮＧＯＩＮＧＢＹＧＯＩＮＧＵＰ；行１９）。下りポートが設定された後、ルーティングメカニズムは、ツリーを下降していくことによってすべての接続された下りスイッチ上において、ＶＭに向かうルートのために上りポートを割当てることができる（ポートについての対応する上り重み（upward weight）を更新する）（ＲＯＵＴＥＵＰＧＯＩＮＧＢＹＧＯＩＮＧＤＯＷＮ；行２０）。次いで、当該プロセスはツリーにおける次のレベルまで上っていくことによって繰返される。すべてのＶＭがルーティングされると、（擬似コードに図示されない）トポロジーにおいてｖＳｗｉｔｃｈ経路とｖＳｗｉｔｃｈ経路との間でバランスを取るように等しい重み付けがなされているにも関わらず、アルゴリズムはまた、ＶＭと同じ方法でｖＳｗｉｔｃｈの物理的ＬＩＤをルーティングする。これは、最小限の再構成方法がライブマイグレーションの文脈において用いられる際にバランスを取るのを向上させるのに望ましい。また、ｖＳｗｉｔｃｈのベースとなる物理的ＬＩＤ上のルーティング経路は、再構成を必要とすることなく、新しいＶＭを迅速にデプロイするために予め定められた経路として用いることができる。しかしながら、一定の期間にわたって、全体的なルーティング性能は、元のｖＳｗｉｔｃｈＦａｔＴｒｅｅルーティングの間にわずかに減少するだろう。性能の低下を制限するために、ある性能しきい値を超えたとき、ｖＳｗｉｔｃｈＦａｔＴｒｅｅに基づいた再構成をオフラインで実行してもよい。

【０１００】

一実施形態に従うと、上述のルーティングメカニズムは、正規の／レガシーなルーティングメカニズムに勝るさまざまな改善を提供することができる。トポロジーにおけるｖＳｗｉｔｃｈまたはＶＭを考慮に入れていない当初のファットツリールーティングアルゴリズムとは異なり、ｖＳｗｉｔｃｈＦａｔＴｒｅｅは、ｖＳｗｉｔｃｈに印付けをして、ｖＳｗｉｔｃｈに接続された他のＶＭからは独立して各々のＶＭをルーティングする。同様に、ｖＳｗｉｔｃｈ間で不均一なＶＭ分配を行なうために、各々のＶＭには、ｖＳｗｉｔｃｈ上で割付けられているリンクの割合に対応する重みが割当てられている。重みは、ファットツリーにおける経路配分のバランスを取るためのポートカウンタを維持するのに用いられる。スキームはまた、一般化された重み付けされたファットツリールーティングを可能にする。この場合、各々のＶＭには、ネットワークにおけるそのトラヒックプロファイルまたは役割の優先順位に基づいて重みを割当てることができる。

【０１０１】

図１１から図１４は、一実施形態に従った、無損失相互接続ネットワークにおいて効率的な仮想化をサポートする例を示す。具体的には、図１１は、４つのスイッチとして、ルートスイッチ９２５および９２６、リーフスイッチ９２０および９２１、さらには、４つの仮想スイッチＶＳ１　９３１、ＶＳ２　９４１、ＶＳ３　９５１およびＶＳ４　９６１を備えた２レベルのファットツリートポロジーを示す。４つの仮想スイッチＶＳ１　９３１、ＶＳ２　９４１、ＶＳ３　９５１およびＶＳ４　９６１には、それぞれ、４つのホスト／ハイパーバイザ９３０、９４０、９５０、９６０が関連付けられており、この場合、４つの仮想スイッチは、８つの仮想マシンＶＭ１　９３２、ＶＭ２　９３３、ＶＭ３　９４２、ＶＭ４　９４３、ＶＭ５　９５２、ＶＭ６　９５３、ＶＭ７　９５４、およびＶＭ８　９６２のために接続性を提供する。

【０１０２】

ｖＳｗｉｔｃｈＦａｔＴｒｅｅルーティングをさらに詳しく説明するために、図１１に示されるように、４つのエンドノード（ｖＳｗｉｔｃｈ）を備えた単純な仮想化されたファットツリートポロジーについて検討する。リーフスイッチ９２０、ＶＳ１およびＶＳ２に接続されたｖＳｗｉｔｃｈの各々は、実行中の２つのＶＭ（ＶＳ１についてはＶＭ１およびＶＭ２、ならびにＶＳ２についてはＶＭ３およびＶＭ４）を有する。第２のリーフスイッチ９２１は、３つのＶＭ（ＶＭ５、ＶＭ６、ＶＭ７）を備えたＶＳ３を有し、１つのＶＭがホストｖＳｗｉｔｃｈ　ＶＳ４の上で実行中である。各々のリーフスイッチは、両方のルートスイッチ９２５および９２６に接続されているため、ルートを介して各々のＶＭに向かうルートを設定するのに利用可能な２つの代替経路が存在している。ＶＳ１に接続されたＶＭのためのルーティングは、ルートスイッチからの選択された下り経路を示す円を用いて、図１２に示される。ＶＭ１は９２５→９２０を用いてルーティングされ、ＶＭ２は９２６→９２０からルーティングされている。対応する下り負荷カウンタは、選択されたリンク上で更新されて、各々のＶＭのために０．５を追加する。同様に、図１３に示されるように、ＶＳ２のためのルートを追加した後、ＶＭ３およびＶＭ４は、リンク９２５→９２０およびリンク９２６→９２０を介してそれぞれルーティングされる。リーフスイッチ９２０に接続されたすべてのＶＭをルーティングした後、たとえＶＭが個々にルーティングされていたとしても、両方のリンク上の下り負荷の合計が等しくなることに留意されたい。リーフスイッチ９２１に接続されたｖＳｗｉｔｃｈ上のＶＭ分配は異なっており、このため、１つのＶＭを備えたｖＳｗｉｔｃｈ（ＶＳ４）は最初にルーティングされることとなるだろう。ルート９２５→９２１がＶＭ８に割付けられ、ＶＳ３に接続された３つのすべてのＶＭが９２６→９２１からルーティングされて、両方の下りリンク上で累積された負荷のバランスが取られた状態を維持するようにする。図１４に示される最後のルーティングでは、トポロジーにＶＭ分配がなされていると想定して、可能な限り、ＶＭに向かう独立したルートと共に、各々のリンク上で負荷のバランスが取られている。

【０１０３】

仮想マシンライブマイグレーション上での最小限のオーバーヘッド再構成

一実施形態に従うと、ＩｔＲＣ（Iterative Reconfiguration：反復再構成）と略され得る動的な再構成メカニズムは、ＶＭがマイグレートされたときに、必要に応じて、ルートの切替えおよび更新をすべてを繰返す。しかしながら、サブネットにおける既存のＬＦＴ（すなわち、既に計算されたＬＦＴであって、サブネット内の各スイッチに存在しているＬＦＴ）に応じて、スイッチのサブセットだけを実際に更新する必要がある。

【０１０４】

図１５は、一実施形態に従った潜在的な仮想マシンマイグレーションを示す。より具体的には、図１５は、ネットワークトポロジーにもかかわらず、対応するリーフスイッチだけがＬＦＴ更新を必要としているリーフスイッチ内のＶＭのマイグレーションの特別な事例を示している。

【０１０５】

図１５に示されるように、サブネットは、いくつかのスイッチ、すなわち、スイッチ１　１３０１～スイッチ１２　１３１２を含み得る。これらのスイッチのうちのいくつかは、スイッチ１　１３０１、スイッチ２　１３０２、スイッチ１１　１３１１、スイッチ１２　１３１２などのリーフスイッチを含み得る。サブネットは、付加的に、いくつかのホスト／ハイパーバイザ１３３０、１３４０、１３５０および１３６０、いくつかの仮想スイッチＶＳ１　１３３１、ＶＳ２　１３４１、ＶＳ３　１３５１およびＶＳ４　１３６１を含み得る。さまざまなホスト／ハイパーバイザは、仮想機能を介して、ＶＭ１　１３３２、ＶＭ２　１３３３、ＶＭ３　１３３４、ＶＭ４　１３４２、ＶＭ５　１３４３およびＶＭ６　１３５２などのサブネット内の仮想マシンをホストすることができる。

【０１０６】

一実施形態に従うと、ＶＭ３が（太字矢印によって示されるように）付随しているハイパーバイザ１３３０からハイパーバイザ１３４０における自由な仮想機能にマイグレートする場合、リーフスイッチ１　１３０１におけるＬＦＴだけが更新される必要がある。な

ぜなら、両方のハイパーバイザが同じリーフスイッチに接続されており、局所的な変更がネットワークの残りの部分に影響を及ぼさないからである。たとえば、最初のルーティングアルゴリズムは、ハイパーバイザ１３６０からハイパーバイザ１３３０に向かうトラフィックが実線（すなわち、１２→９→５→３→１）によって印付けされた第１の経路を追従すると判断する。同様に、ハイパーバイザ１３６０からハイパーバイザ１３４０に向かうトラフィックは、破線（すなわち、１２→１０→６→４→１）によって印付けされた第２の経路を追従する。ＶＭ３がマイグレートされ、ネットワークを再構成するためにＩｔＲＣが用いられる場合、ＶＭ３に向かうトラフィックは、マイグレーションの前にハイパーバイザ１３３０に向かう第１の経路を追従し、マイグレーションの後、ハイパーバイザ１３４０に向かう第２の経路を追従することとなるだろう。この状況においては、ファットツリールーティングアルゴリズムが最初のルーティングのために用いられたと想定すると、ＩｔＲＣ法は、スイッチの総数の半分（６／１２）を更新するだろう。しかしながら、マイグレートされたＶＭを接続されたままにしておくためにリーフスイッチを１つだけ更新する必要がある。

【０１０７】

一実施形態に従うと、ＶＭマイグレーションの後にスイッチ更新の回数を制限することによって、ネットワークをより速く再構成することができ、従来のルーティング更新の際に必要とされる時間およびオーバーヘッドを減らすことができる。これは、トポロジーに依存しないスカイライン技術（topology-agnostic skyline technique）に基づいて、ＦＴｒｅｅＭｉｎＲＣと称される、ファットツリー上でのＶＭマイグレーションをサポートするためのトポロジー認識型高速再構成方法によって達成することができる。

【０１０８】

ファットツリーにおけるサブツリーおよびスイッチタプル

一実施形態に従うと、以下の記述は、例示的なファットツリーネットワークとしてＸＧＦＴを用いて、最小限のオーバーヘッドネットワーク再構成方法であるＦＴｒｅｅＭｉｎＲＣを利用する。しかしながら、ここで提示される概念は、ＰＧＦＴおよびＲＬＦＴにとっても有効である。ＸＧＦＴ（ｎ；m_1，．．．，m_n；w_1，．．．，w_n）は、ｎ＋１レベルのノードを備えたファットツリーである。レベルは０からｎで表わされ、計算ノードがレベルｎにあり、スイッチが他のすべてのレベルにある。子がない計算ノードを除いては、レベルｉ、０≦ｉ≦ｎ－１におけるすべてのノードは、m_iの子ノードを有する。同様に、親がないルートスイッチを除いては、レベルｉ、１≦ｉ≦ｎにおける他のすべてのノードはw_i＋１の親ノードを有する。

【０１０９】

【数２】

ＸＧＦＴ（ｎ＋１；m_1，．．．，m_{n+1}；w_1，．．．，w_{n+1}）は、新しい最上位レベルにおける$\prod_{i=0}^{n+1} w_i$の付加的スイッチと、ＸＧＦＴ（ｎ＋１；m_1，．．．，m_n；w_1，．．．，w_n）のm_nの別個のコピーを接続することによって再帰的に構築される。この定義を用いることにより、以下の特性が適用される：ｎ＞０の場合、ｎ＋１レベルである各々のＸＧＦＴは、m_nサブツリーから構成される（すなわち、ｌレベルであるＸＧＦＴにおけるｎレベルの各サブツリーの場合（ｌ＞ｎ）、ｎ＋１レベルである１つのイミディエイト・スーパーツリーが存在する。これはm_nのｎレベルサブツリーを接続するものである）。同様に、ネットワーク接続性の観点から、ＸＧＦＴにおける各々のサブツリーを別個のＸＧＦＴと見なすことができ、サブツリーにおける最上位レベルのスイッチがそのイミディエイト・スーパーツリーに向かうそのスカイラインを定義する。

【０１１０】

一実施形態に従うと、ｎ＋１レベルであるＸＧＦＴにおける各々のスイッチは固有のｎ

タプル（ｌ，ｘ１，ｘ２，．．．，ｘｎ）によって表わすことができる。左端のタプル値（ｌ）はツリーが位置するレベルを表わしており、残りの値（ｘ１，ｘ２，．．．，ｘｎ）は、他のスイッチに対応するツリーにおけるスイッチの位置を表わしている。特に、レベルｌにおけるスイッチＡ（ｌ，ａ１，．．．，ａｌ，．．．，ａｎ）は、ｉ＝ｌ＋１である場合を除いて、すべての値についてａｉ＝ｂｉであるとき、かつそのときに限り、レベルｌ＋１，（ｌ＋１，ｂ１，．．．，ｂｌ，ｂｌ＋１，．．．，ｂｎ）におけるスイッチＢに接続される。

【０１１１】

図１６は、一実施形態に従ったスイッチタプルを示す。より具体的には、当該図は、例示的なファットツリーであるＸＧＦＴ（４；２，２，２，２；２，２，２，１）のために実現されたＯｐｅｎＳＭのファットツリールーティングアルゴリズムによって割付けられるようなスイッチタプルを示している。ファットツリー１４００は、スイッチ１４０１～１４０８、１４１１～１４１８、１４２１～１４２８および１４３１～１４３８を含み得る。ファットツリーが（リーフレベルにおける列３まで、ルートレベルにおける列０として印付けされた）ｎ＝４のスイッチレベルを有しているので、ファットツリーは、各々がｎ′＝ｎ－１＝３スイッチレベルであるｍ１＝２の第１レベルサブツリーで構成されている。これは、図において、レベル１から３までのスイッチを囲んでいる破線によって規定される２つのボックスによって示されている。各々の第１レベルのサブツリーが０または１の識別子を受取る。第１レベルのサブツリーの各々は、各々がリーフスイッチを上回っているｎ″＝ｎ′－１＝２のスイッチレベルであるｍ２＝２の第２レベルのサブツリーから構成されている。これは、図において、レベル２から３までのスイッチを囲んでいる点線によって規定される４つのボックスによって示されている。各々の第２レベルのサブツリーは０または１の識別子を受取る。同様に、リーフスイッチの各々は、図において、鎖線によって規定される８つのボックスによって示されるサブツリーと見なすこともできる。これらのサブツリーの各々は０または１の識別子を受取る。

【０１１２】

一実施形態に従うと、図に例示されているように、４つの数字のタプルなどのタプルは、さまざまなスイッチに割当てることができ、タプルの各々の数字は、タプルにおける各々の値の位置についての特定のサブツリー対応を示している。たとえば、（スイッチ１＿３と参照され得る）スイッチ１４１３は、レベル１におけるその位置と０番目の第１レベルのサブツリーとを表わしているタプル１．０．１．１に割当てることができる。

【０１１３】

<u>ライブマイグレーションの文脈におけるＦＴｒｅｅＭｉｎＲＣを用いたファットツリー認識型の最小再構成</u>

一実施形態に従うと、スイッチタプルは、トポロジーにおけるサブツリーに対応するスイッチの位置についての情報を符号化する。ＦＴｒｅｅＭｉｎＲＣは、ライブＶＭマイグレーションの場合における迅速な再構成を可能にするためにこの情報を用いることができる。タプル情報は、ＶＭがマイグレートされたときにＳＭによって再構成される必要のあるスイッチの数が最も少ないスカイランを発見するために用いることができる。特に、ＶＭがファットツリートポロジーにおける２つのハイパーバイザ間でマイグレートされると、更新される必要のある最小数のスイッチを表わしているスカイラインは、マイグレーションに関与しているすべてのサブツリーのうちすべての最上位レベルのスイッチによって形成されている。

【０１１４】

一実施形態に従うと、ＶＭがライブマイグレートされると、スイッチ印付けメカニズムを両方のリーフスイッチから開始することができる。この場合、送信元ハイパーバイザと宛先ハイパーバイザとが接続され、スイッチのタプルを比較する。タプル同士が一致する場合、メカニズムは、ＶＭがリーフスイッチ内でマイグレートされていると判断することができる。これにより、再構成のために対応するリーフスイッチだけに印が付けられる。しかしながら、タプルが一致していなければ、送信元リーフスイッチおよび宛先リーフス

イッチの両方からの上りリンクがトレースされる。１レベル上に位置するスイッチは、リーフレベルのサブツリーが接続されているイミディエイト・スーパーツリーのうち最上位レベルのスイッチであり、ツリーを下方へと横切る際にリーフスイッチに到達する前に生じる可能性のある唯一のホップである。次いで、当該メカニズムは、送信元リーフスイッチタプルおよび宛先リーフスイッチタプルを新しくトレースされたスイッチと比較することができ、その時点のレベルを反映させるためにタプル値を調整した後、その時点のツリーのサブツリーに対応する値がワイルドカードにされる。さらに、（対応するサブツリーのための最上位レベルのスイッチである）トレースされたスイッチは更新されるべく印付けされ、送信元スイッチタプルおよび宛先スイッチタプルの両方からの比較がトレースされたすべてのスイッチのタプルと一致する場合、トレースが停止される。他の場合には、メカニズムが両端から共通の先祖スイッチを特定するまで、同じ手順が繰返される。最悪の場合、ファットツリートポロジーのルートスイッチに到達した後、メカニズムを停止することができる。すべての上り経路のトレースがリーフレベルから開始されており、かつ、連続したサブツリーのスカイラインスイッチに印付けされているので、メカニズムがマイグレーションによって影響される最上位のサブツリーに到達した場合、当該メカニズムは、その途中で、下位レベルスイッチに向かう潜在的なトラフィックゲートウェイであるすべてのスイッチや、ライブマイグレーションに関与するハイパーバイザを既に選択してしまっている。これにより、当該メカニズムは、ネットワークのうちライブマイグレーションによって影響を受けた部分のスカイラインを形成するすべてのスイッチに印を付けた。

【０１１５】

一実施形態に従うと、スイッチ印付けメカニズムは、物理的接続の観点から更新される必要のある、最小数のスイッチを発見する。しかしながら、これらのスイッチのすべてが再構成によって影響を受けたＬＩＤに対するルーティングアルゴリズムによって計算されたアクティブな経路を含むとは限らない可能性もある。このため、アクティブなルートを含んでいるスイッチには更新手順において優先順位が付けられる一方で、スイッチのうち二次ルートを有する残りのスイッチは後で更新することができる。

【０１１６】

一実施形態に従うと、ファットツリールーティングメカニズムは、常に、同じルートスイッチを介して所与の宛先にトラフィックをルーティングする。トポロジーにおいてはルートスイッチとエンドノードとの間に単一の経路だけが存在しているので、所与のエンドノードを表わすために選択されたルートスイッチが位置特定されると、エンドノードにトラフィックをルーティングするために用いられる中間スイッチを見出すことができる。アクティブなルートを発見するために、経路は、関与するハイパーバイザの送信元ＬＩＤから宛先ＬＩＤにまで、またはこれの逆の態様でトレースすることができる。再構成のために既に選択されていたスイッチのサブセットであるスイッチに印を付けることができ、それらのスイッチのＬＦＴ更新に優先順位を付けることができる。その後、すべてのＬＦＴを有効に維持するために、残りの選択されたスイッチを更新することができる。

【０１１７】

図１７は、一実施形態に従った再構成プロセスを示す。ファットツリー１４００は、スイッチ１４０１～１４０８、１４１１～１４１８、１４２１～１４２８および１４３１～１４３８を含み得る。ファットツリーが（リーフレベルにおける列３まで、ルートレベルにおける列０として印付けされた）ｎ＝４スイッチレベルを有しているので、ファットツリーは、各々がｎ′＝ｎ－１＝３のスイッチレベルであるm_1＝２の第１レベルのサブツリーから構成されている。これら第１レベルのサブツリーの各々は、各々がリーフスイッチを上回っているｎ″＝ｎ′－１＝２のスイッチレベルであるm_2＝２の第２レベルのサブツリーから構成されている。同様に、リーフスイッチの各々もサブツリーと見なすことができる。

【０１１８】

一実施形態に従うと、図１７は、タプル３．０．０．０および３．０．１．１を備えたリーフスイッチに接続された２つのハイパーバイザ間でＶＭがマイグレートされている状

況を示す。これらの２つのタプルは、選択されたリーフスイッチから上方向への経路をメカニズムがトレースする際の比較についての基準として用いられる。この例においては、共通の先祖スイッチがレベル１上で発見される。レベル０はルートレベルであり、レベル３はリーフレベルである。表示されたタプル情報を有するスイッチ間のリンクはメカニズムの実行中ずっとトレースすることができるリンクであり、それらの同じスイッチすべてに更新するための印を付けることができる。強調表示された５つのスイッチ（スイッチ１４３１、１４２１、１４１１、１４２３および１４３４）およびそれらの間のリンクは、アクティブなルートを表わしており、それらのＬＦＴ更新に優先順位を付けることができる。

【０１１９】

一実施形態に従うと、ライブマイグレーションをサポートする仮想化されたデータセンタにおける最小限のオーバーヘッドに迅速な接続性を提供するために、ＦＴｒｅｅＭｉｎＲＣは、スイッチに送信される必要のあるＬＦＴ更新の回数を最小限にする。

【０１２０】

図１８は、一実施形態に従った、無損失相互接続ネットワークにおける効率的な仮想化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１８１０において、当該方法は、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、少なくともリーフスイッチを含む１つ以上のスイッチを設けることができ、当該１つ以上のスイッチの各々は複数のポートを含み、さらに、複数のホストチャネルアダプタを設けることができ、ホストチャネルアダプタの各々は少なくとも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は１つ以上のスイッチを介して相互接続されており、さらに、複数のハイパーバイザを設けることができ、複数の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ており、さらに、複数の仮想マシンを設けることができ、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。

【０１２１】

ステップ１８２０において、当該方法は、予めポピュレートされたローカル識別子（local identifier：ＬＩＤ）アーキテクチャを備えた仮想スイッチまたは動的ＬＩＤ割当てアーキテクチャを備えた仮想スイッチのうち１つ以上を備えた複数のホストチャネルアダプタを配置することができる。

【０１２２】

ステップ１８３０において、当該方法は、各々の仮想スイッチにＬＩＤを割当てることができ、割当てられたＬＩＤは関連付けられた物理機能のＬＩＤに対応している。

【０１２３】

ステップ１８４０において、当該方法は、仮想スイッチの各々に割当てられたＬＩＤに少なくとも基づいて１つ以上のリニアフォワーディングテーブルを計算することができ、１つ以上のＬＦＴの各々は、１つ以上のスイッチのうちの一のスイッチに関連付けられている。

【０１２４】

図１９は、一実施形態に従った、無損失相互接続ネットワークにおける効率的な仮想化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１９１０において、当該方法は、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、１つ以上のマイクロプロセッサと、１つ以上のスイッチとを設けることができ、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ポートを含み、さらに、複数のホストチャネルアダプタを設けることができ、ホストチャネルアダプタの各々は少なくとも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は１つ以上のスイッチを介して相互接続されており、さらに、複数のハイパーバイザを設けることができ、複数の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ており、さらに、複数の仮想マシンを設けることができ、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。

【０１２５】

ステップ１９２０において、当該方法は、予めポピュレートされたローカル識別子（ＬＩＤ）アーキテクチャを備えた仮想スイッチまたは動的ＬＩＤ割当てアーキテクチャを備えた仮想スイッチのうち１つ以上を備えた複数のホストチャネルアダプタを配置することができる。

【０１２６】

ステップ１９３０において、当該方法は、仮想スイッチの各々に複数のｐＬＩＤのうち１つのｐＬＩＤを割当てることができる。割当てられたｐＬＩＤは関連付けられた物理機能のｐＬＩＤに対応している。

【０１２７】

ステップ１９４０において、当該方法は、複数の仮想マシンの各々に複数のｖＬＩＤのうち１つのｖＬＩＤを割当てることができ、ＬＩＤスペースは、複数のｐＬＩＤおよび複数のｖＬＩＤを含む。

【０１２８】

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。

【０１２９】

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。

【０１３０】

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。

【０１３１】

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 integrated 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。

【０１３２】

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。

【０１３３】

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。

【０１３４】

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。

【０１３５】

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。

【図面】

【図１】

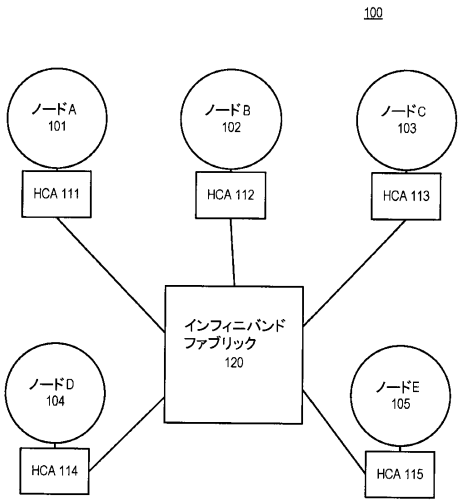

FIGURE 1

【図２】

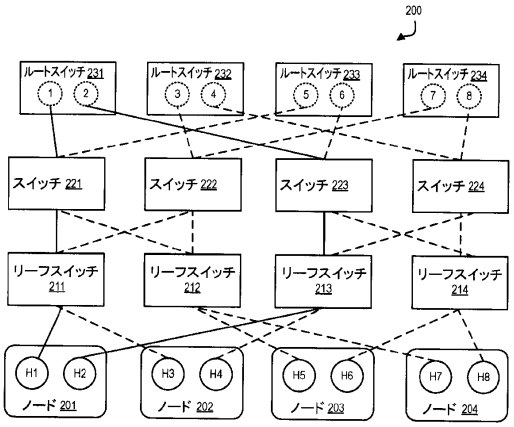

FIGURE 2

【図３】

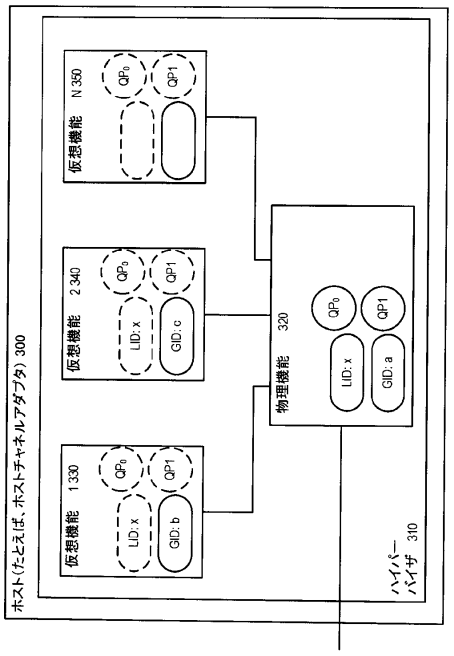

FIGURE 3

【図４】

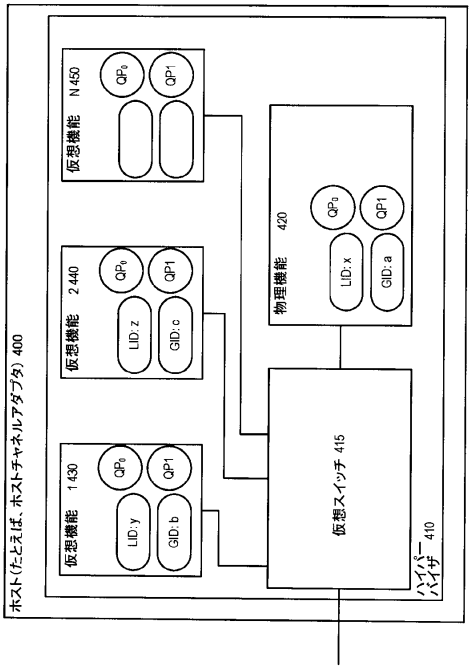

FIGURE 4

【図５】

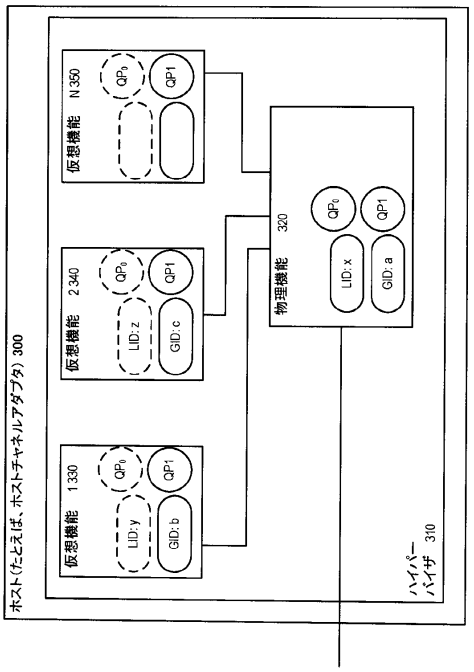

FIGURE 5

【図６】

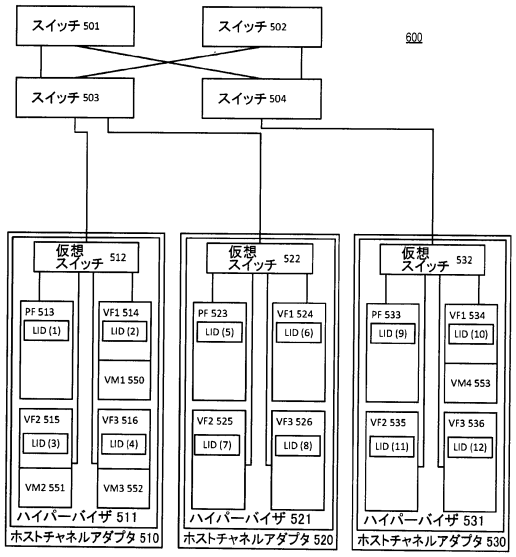

FIGURE 6

【図７】

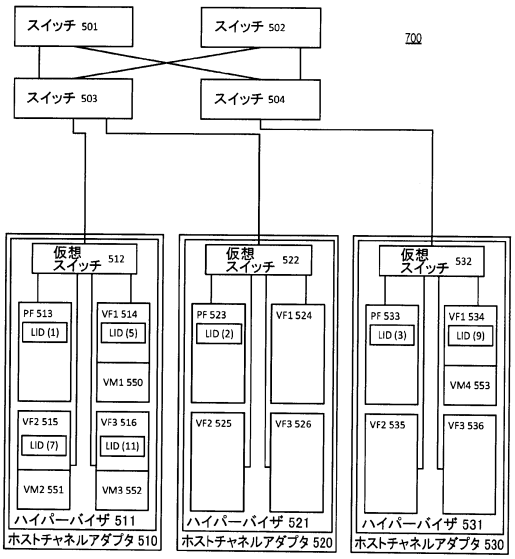

FIGURE 7

【図８】

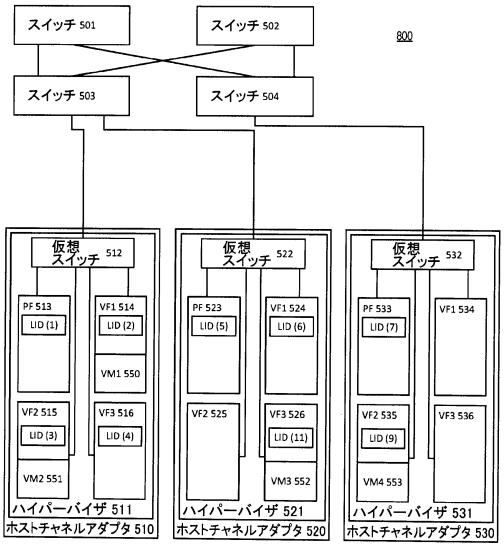

FIGURE 8

【図９】

900	仮想レーン（VL）－4ビット
901	リンクバージョン（Lver）－4ビット
902	サービスレベル（SL）－4ビット
903	LID拡張フラグ（LEXTF）－1ビット
904	予備（R1）－1ビット－ゼロ
905	リンク次ヘッダ（LNH）－2ビット
906	宛先ローカルID（DLID）－16ビット
907	DLIDプレフィックス拡張（DPF）－2ビット
908	SLIDプレフィックス拡張（SPF）－2ビット
909	予備（R2）－1ビット－ゼロ
910	パケット長（PktLen）－11ビット
911	送信元ローカルID（SLID）－16ビット

FIGURE 9

【図１０】

912	16ビットDLID＝0によって索引付けされたエントリ0（標準IBポート番号を含む）
	... 916
913	16ビットDLID＝48K－1によって索引付けされたエントリ48K－1（IBポート番号を含む）

914	18ビットDPF＋DLID＝64Kによって索引付けされたエントリ0（標準IBポート番号を含む）
	... 917
915	18ビットDPF＋DLID＝256K－1によって索引付けされたエントリ256K－1（標準IBポート番号を含む）

FIGURE 10

【図１１】

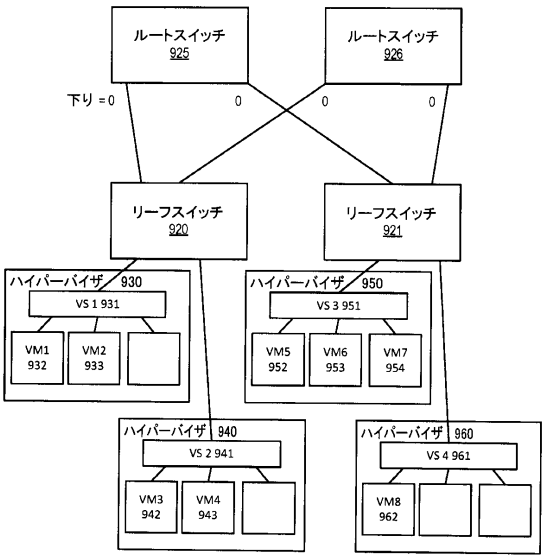

FIGURE 11

【図１２】

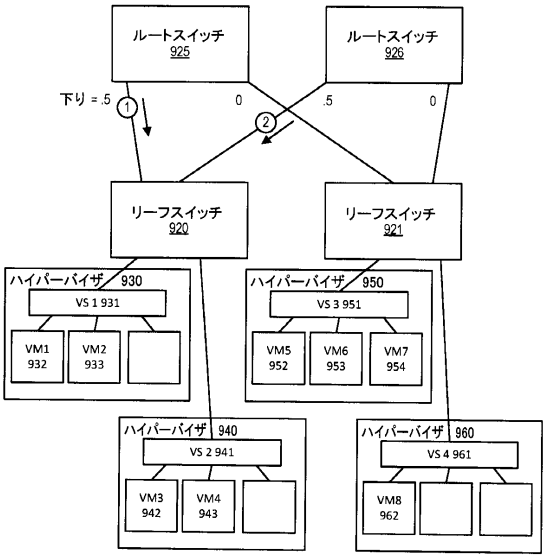

FIGURE 12

【図１３】

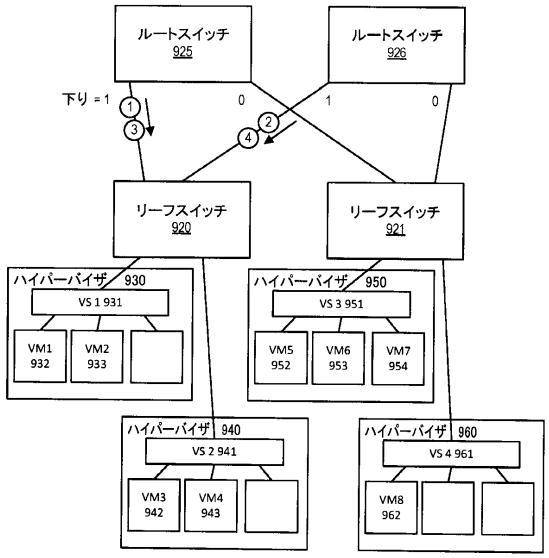

FIGURE 13

【図１４】

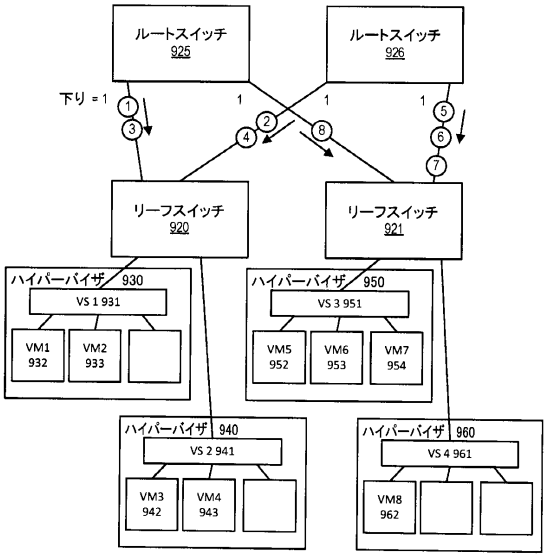

FIGURE 14

【図１５】

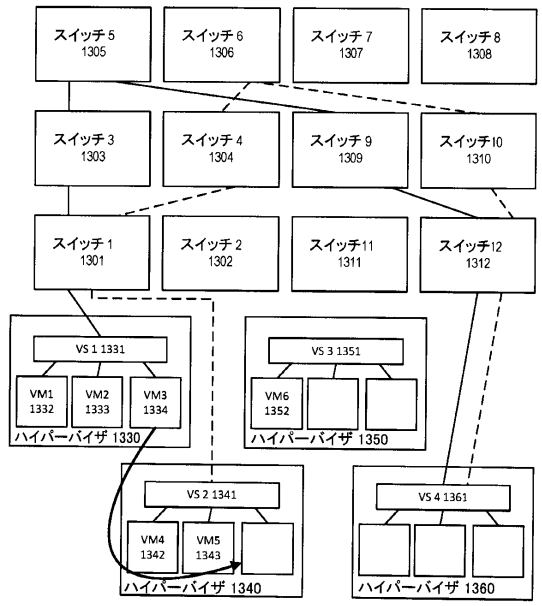

FIGURE 15

【図１６】

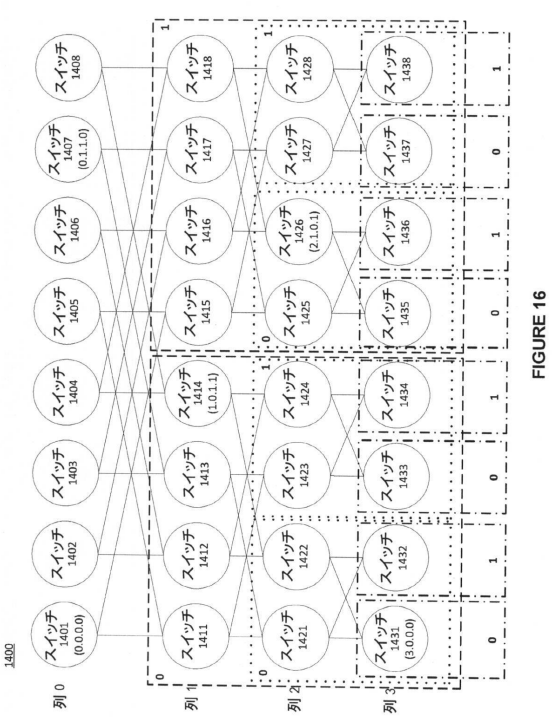

FIGURE 16

【図１７】

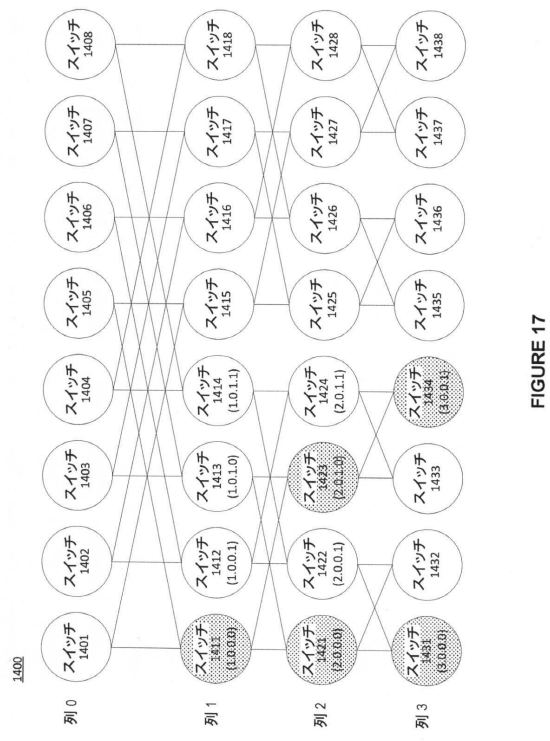

FIGURE 17

【図１８】

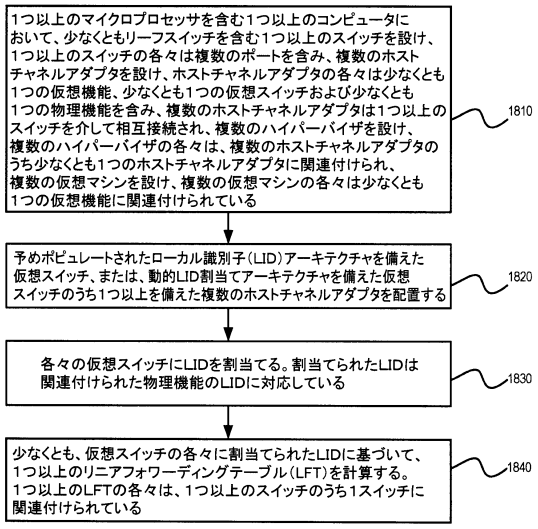

FIGURE 18

【図１９】

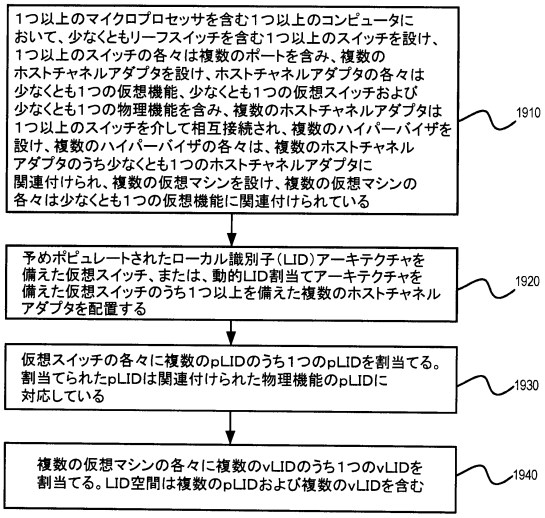

FIGURE 19

フロントページの続き

米国(US)
(31)優先権主張番号　62/261,103
(32)優先日　平成27年11月30日(2015.11.30)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　15/210,595
(32)優先日　平成28年7月14日(2016.7.14)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　15/210,599
(32)優先日　平成28年7月14日(2016.7.14)
(33)優先権主張国・地域又は機関
米国(US)
ルクグレンダ、９
(72)発明者　グラン，アーンスト・ガンナー
ノルウェー、１３２５　リュサケール、ピィ・オゥ・ボックス・１３４
審査官　漆原　孝治
(56)参考文献　特開２０１１－０２８４０８（ＪＰ，Ａ）
特開２０１３－０６９２６０（ＪＰ，Ａ）
Evangelos Tasoulas，Towards the InfiniBand SR-IOV vSwitch Architecture，2015 IEEE INTERNATIONAL CONFERENCE ON CLUSTER COMPUTING，2015年09月08日，pp.371-380
(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１５／１７３
Ｇ０６Ｆ　９／４５５
Ｇ０６Ｆ　９／５０
Ｈ０４Ｌ　６１／００
Ｈ０４Ｌ　６１／１０