(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6492080号
(P6492080)
(45)発行日　平成31年3月27日(2019.3.27)　(24)登録日　平成31年3月8日(2019.3.8)

(51)Int.Cl.　F I
H04L 12/44 (2006.01)　H04L 12/44 Z

請求項の数 13　(全 18 頁)

(21)出願番号	特願2016-531743 (P2016-531743)
(86)(22)出願日	平成26年7月18日 (2014.7.18)
(65)公表番号	特表2016-525853 (P2016-525853A)
(43)公表日	平成28年8月25日 (2016.8.25)
(86)国際出願番号	PCT/US2014/047280
(87)国際公開番号	WO2015/017145
(87)国際公開日	平成27年2月5日 (2015.2.5)
審査請求日	平成29年6月21日 (2017.6.21)
(31)優先権主張番号	61/859,650
(32)優先日	平成25年7月29日 (2013.7.29)
(33)優先権主張国	米国(US)
(31)優先権主張番号	14/226,288
(32)優先日	平成26年3月26日 (2014.3.26)
(33)優先権主張国	米国(US)

(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
特許業務法人深見特許事務所
(72)発明者　ボグダンスキー，バルトシュ
ノルウェー、エヌ－０２７５　オスロ、エイチ・０２０３、ホフ・テラス、１５
(72)発明者　ヨンセン，ビョルン・ダグ
ノルウェー、エヌ－０６８７　オスロ、ビルベルクグレンダ、９

最終頁に続く

(54)【発明の名称】ミドルウェアマシン環境におけるマルチホーム型ファットツリー・ルーティングをサポートするためのシステムおよび方法

(57)【特許請求の範囲】

【請求項１】

ネットワーク環境のサブネットにおいてマルチホーム型ルーティングをサポートするための方法であって、

前記サブネットにおいてリーフスイッチ上のスイッチポートに関連付けられたエンドノードを提供するステップを含み、前記エンドノードは複数のポートを有し、前記方法はさらに、

前記エンドノードが有する各々の前記ポートのためにルーティングを実行するステップと、

前記ルーティングのために、前記エンドノードが有する前記複数のポートが相互に独立した経路を取ることを保証するステップとを含む、方法。

【請求項２】

前記エンドノードが有する前記複数のポートは、第１ポートおよび第２ポートを含み、

前記サブネットは、複数の前記エンドノードと、前記複数のエンドノードに接続される複数の前記リーフスイッチと、前記複数のリーフスイッチに接続される複数のルートスイッチとを備え、前記第１ポートおよび前記第２ポートのそれぞれは、前記複数のリーフスイッチのうちの異なるリーフスイッチに接続され、

前記複数のルートスイッチ、前記複数のリーフスイッチおよび前記複数のエンドノードの間の接続はファットツリー・トポロジーに基づいている、請求項１に記載の方法。

【請求項３】

前記エンドノードは、複数のポートを介して前記ファットツリー・トポロジーの２つ以上の部分に接続されるマルチホーム型ノードである、請求項２に記載の方法。
【請求項４】
前記エンドノードの前記複数のポートにおける１つのポートに関連付けられた経路上の各々のスイッチにマーク付けするステップと、
前記エンドノードの前記複数のポートにおける別のポートに関連付けられた別の経路が、前記マーク付けされたスイッチを用いることを防止するステップとをさらに含む、請求項１～３のいずれか１項に記載の方法。
【請求項５】
冗長なスイッチがない場合に、前記エンドノードの前記複数のポートにおける別のポートに関連付けられた別の経路が、１つ以上のマーク付けされたスイッチを用いることを可能にするステップと、
相互に独立したリンクが前記１つ以上のマーク付けされたスイッチ上に存在する場合に、同じエンドノードの複数のポートのうちの異なる目標ポートのための独立したリンクを選択するステップとをさらに含む、請求項４に記載の方法。
【請求項６】
前記エンドノードの前記複数のポートの前記ルーティングを完了した後に各々のマーク付けされたスイッチのマーク付けを解除するステップをさらに含む、請求項４または５に記載の方法。
【請求項７】
前記エンドノードからの相互に独立した各々の経路を異なるスパインスイッチに関連付けるステップをさらに含む、請求項１～６のいずれか１項に記載の方法。
【請求項８】
前記エンドノードの前記複数のポートの前記ルーティングを完了した後に、前記エンドノードをルーティングされたエンドノードとしてマーク付けするステップをさらに含む、請求項１～７のいずれか１項に記載の方法。
【請求項９】
前記エンドノードが別のリーフスイッチに遭遇した場合に、前記エンドノードが再びルーティングされることを防止するステップをさらに含む、請求項８に記載の方法。
【請求項１０】
前記エンドノードをパラメータとして採用するルーティングアルゴリズムを用いるステップをさらに含む、請求項１～９のいずれか１項に記載の方法。
【請求項１１】
システムによって実行されたときに請求項１～１０のいずれか１項に記載の方法を前記システムに実行させる命令を含むコンピュータプログラム。
【請求項１２】
命令を有するコンピュータプログラムであって、前記命令が実行されると、
ネットワーク環境のサブネットにおいてリーフスイッチ上のスイッチポートに関連付けられ、複数のポートを有するエンドノードを提供するステップと、
前記エンドノードの各々の前記ポートのためにルーティングを実行するステップと、
前記エンドノードの前記複数のポートが相互に独立した経路を取ることを保証するステップとをシステムに実行させる、コンピュータプログラム。
【請求項１３】
ネットワーク環境のサブネットにおいてマルチホーム型ルーティング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、
前記１つ以上のマイクロプロセッサ上で実行されるサブネットマネージャとを含み、前記サブネットマネージャは、
前記サブネットにおいてリーフスイッチ上のスイッチポートに、複数のポートを有するエンドノードを関連付け、

前記エンドノードの各々の前記ポートのためにルーティングを実行し、
前記エンドノードの前記複数のポートが前記ルーティングのために相互に独立した経路を取ることを保証するように動作する、システム。

【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題材を含んでいる。著作権の所有者は、特許商標庁の包袋または記録に掲載されるように特許文献または特許情報開示を誰でも複製できることに対して異議はないが、その他の点ではすべての如何なる著作権をも保有する。

【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、特にネットワーク環境に関する。

【背景技術】

【０００３】

背景：

ファットツリー・トポロジーは、高性能コンピューティング（ＨＰＣ：high performance computing）クラスタと、インフィニバンド（ＩＢ：InfiniBand）技術に基づいたクラスタとのために用いられる。たとえば、ファットツリー・トポロジーは、ＭｉｌｋｙＷａｙ－２などの最速のスーパーコンピュータにおいて用いられる。また、ファットツリーＩＢシステムは、Ｓｔａｍｐｅｄｅ、ＴＧＣＣ　ＣｕｒｉｅおよびＳｕｐｅｒＭＵＣなどの大型の設備を含む。

【０００４】

これらは、概して、本発明の実施形態が対処するように意図された分野である。

【発明の概要】

【課題を解決するための手段】

【０００５】

概要：

この明細書中に記載されるシステムおよび方法は、ファットツリーまたは同様のトポロジーを用いるインフィニバンドアーキテクチャに基づき得るネットワーク環境において、マルチホーム型ルーティングをサポートすることができる。システムは、ネットワークファブリックにおいてリーフスイッチ上のスイッチポートに関連付けられたエンドノードを提供することができる。さらに、システムは、エンドノード上の複数のポートの各々のためにルーティングを実行し、エンドノード上の複数のポートが相互に独立した経路を取ることを保証することができる。

【図面の簡単な説明】

【０００６】

【図１】ネットワーク環境におけるファットツリー・ルーティングをサポートする例を示す図である。

【図２】本発明の実施形態に従った、ネットワーク環境におけるマルチホーム型ルーティングをサポートする例を示す図である。

【図３】本発明の実施形態に従った、ネットワーク環境におけるファットツリー・ルーティングをサポートするために冗長性を提供する例を示す図である。

【図４】本発明の実施形態に従った、ネットワーク環境におけるマルチホーム型ルーティングをサポートするための例示的なフローチャートである。

【図５】本発明の実施形態が実現され得るコンピュータシステムを示すブロック図である。

【図６】本発明の実施形態を実現するためのシステムを示すブロック図である。

【発明を実施するための形態】

【０００７】
詳細な説明：
本発明は、添付の図面において、限定ではなく例示のために示されており、図中同様の参照符号は同様の要素を指している。この開示において「或る」、「１つの」または「いくつかの」実施形態と言及する場合、これは必ずしも同じ実施形態である必要はなく、このように言及する場合、少なくとも１つを意味している。
【０００８】
以下の本発明の説明では、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）ネットワークを用いている。他のタイプの高性能ネットワークを制限なく使用できることが当業者にとって明らかになるだろう。また、以下の本発明の説明では、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・トポロジーを用いている。他のタイプのファブリック・トポロジーを制限なく使用できることが当業者にとって明らかになるだろう。
【０００９】
この明細書中に記載されるシステムおよび方法により、ネットワーク環境におけるマルチホーム型ルーティングをサポートすることができる。
【００１０】
インフィニバンドアーキテクチャ
インフィニバンドアーキテクチャ（ＩＢＡ：InfiniBand Architecture）は２層型トポロジー分割をサポートする。下層においては、ＩＢネットワークはサブネットと称される。この場合、サブネットは、スイッチおよび二地点間リンクを用いて相互接続される１組のホストを含み得る。上層においては、ＩＢファブリックは、ルータを用いて相互接続することができる１つ以上のサブネットを構成する。
【００１１】
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：local identifier）を用いてアドレス指定することができ、単一のサブネットは４９１５１ＬＩＤに制限することができる。サブネット内においてのみ有効なローカルアドレスであるＬＩＤに加えて、各々のＩＢデバイスは、その不揮発性メモリに焼きつけられる６４ビットのグローバル固有識別子（ＧＵＩＤ：global unique identifier）を有し得る。ＧＵＩＤは、ＩＢ層３（Ｌ３）アドレスであるグローバル識別子（ＧＩＤ：global identifier）を形成するために用いることができる。ＧＩＤは、ＩＰｖ６のような１２８ビットアドレスを形成するように６４ビットサブネット識別子（ＩＤ：identifier）を６４ビットＧＵＩＤと連結することによって作成することができる。たとえば、さまざまなポートＧＵＩＤを、ＩＢファブリックに接続されたポートに割当てることができる。
【００１２】
加えて、サブネットマネージャ（ＳＭ：subnet manager）は、ＩＢファブリックにおけるルーティングテーブル計算を実行する役割を果たすことができる。ここで、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ソースと宛先対との間において完全な接続性、デッドロック自由性および適切なロードバランシングを獲得することを目的としている。
【００１３】
サブネットマネージャは、ネットワーク初期化時間にルーティングテーブルを計算することができる。さらに、ルーティングテーブルは、最適な性能を保証するために、トポロジーが変化するたびに更新することができる。通常の動作中、サブネットマネージャは、トポロジーの変化をチェックするために、ネットワークに対して周期的な光掃引を実行することができる。光掃引中に変化が発見された場合、または、ネットワークの変化を信号で伝えるメッセージ（トラップ）がサブネットマネージャによって受信された場合、サブネットマネージャは、発見された変化に応じてネットワークを再構成することができる。
【００１４】
たとえば、サブネットマネージャは、たとえば、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、リンクが排除された場合など、ネットワークトポロジーが変化した場合に、ネットワークを再構成することができる。再構成するステップは、ネットワーク初期化中に実行されるステップを含み得る。さらに、再構成には、ネットワークの変化が起こった局所的な範囲が含まれてもよく、この局所的な範囲は、サブネットに制限されている。また、大型のファブリックをルータでセグメント化することにより、再構成範囲が制限される可能性がある。

【００１５】

加えて、ＩＢネットワークは、無損失のネットワーキング技術に基づいたものであって、いくつかの条件下ではデッドロックし易くなる可能性がある。たとえば、デッドロックが起こる可能性があるＩＢネットワークにおいては、バッファまたはチャネルなどのネットワークリソースが共有されており、パケットドロップが許容されていない。ここで、デッドロックを発生させる必須条件として、周期的な信頼依存性の生成が挙げられる。これは、周期的な信頼依存性がデッドロックを発生させる可能性があることを意味している。他方で、これは、周期的な信頼依存性が存在しているときは常にデッドロックが起こるであろうことを意味しているわけではない。

【００１６】

ファットツリー・ルーティング

ファットツリー・トポロジーは、高性能相互接続をサポートするためのさまざまな利点を提供することができる。これらの利点は、デッドロック自由性、固有のフォールトトレランスおよび完全な二分帯域幅を含み得る。デッドロック自由性があれば、デッドロックの回避を特別に考慮することなく、ツリー構造を用いることによってファットツリーをルーティングすることが可能となる。固有のフォールトトレランスがあれば、個々のソース宛先対の間に複数の経路が存在することによりネットワークの障害に対処することが容易になる。完全な二分帯域幅があれば、ネットワークは、二等分にした当該ネットワーク間における通信を最高速度で維持することができる。

【００１７】

さらに、ファットツリー・ルーティングアルゴリズムを用いることにより、基礎をなすファットツリー・トポロジーの効率的な使用をサポートすることができる。以下のアルゴリズム１は、例示的なファットツリー・ルーティングアルゴリズムである。

【００１８】

【表１】

アルゴリズム１　*route_to_cns() 機能*

要求: アドレス指定が完了

保証: すべての *hca_ports* がルーティングされる

```
1: for swleaf = 0 to max_leaf_sw do
2:         for swleaf.port = 0 to max_ports do
3:                 hca_lid = swleaf.port-> remote_lid
4:                 swleaf.routing table[hca_lid] = swleaf.port
5:                 route_downgoing_by_going_ up()
6:         end for
7: end for
```

【００１９】

上に示されるように、ルーティング機能であるｒｏｕｔｅ＿ｔｏ＿ｃｎｓ（）は、一連のリーフスイッチにわたって繰り返すことができる（１～７行目）。選択されたリーフスイッチごとに、ルーティング機能は、たとえば、ポート番号付けシーケンスにおいて、選択されたリーフスイッチに接続される各々のエンドノードポートに向けてルーティングす

ることができる（２～６行目）。
【００２０】
さらに、特定のＬＩＤに関連付けられるエンドノードポートをルーティングする場合、ルーティング機能は、ネットワークトポロジーにおいて１レベル上がって下降経路をルーティングすることができ、各々のスイッチポートをルーティングする場合、ルーティング機能が下がり、上昇経路をルーティングすることができる。このプロセスは、ルートスイッチレベルに達するまで繰り返すことができる。その後、すべてのノードに向かう経路がルーティングされ、ファブリックにおけるすべてのスイッチのリニアフォワーディングテーブル（ＬＦＴ：linear forwarding table）に挿入される。
【００２１】
たとえば、ｒｏｕｔｅ＿ｄｏｗｎｇｏｉｎｇ＿ｂｙ＿ｇｏｉｎｇ＿ｕｐ（）機能（５行目）は反復機能であってもよく、この反復機能は、経路のバランスをとることができ、ｒｏｕｔｅ＿ｕｐｇｏｉｎｇ＿ｂｙ　ｇｏｉｎｇ＿ｄｏｗｎ（）機能を呼び出し、この機能が、ファットツリーにおける上り経路をスイッチを介してルーティングするが、このルーティングの宛先からは、ｒｏｕｔｅ＿ｄｏｗｎｇｏｉｎｇ＿ｂｙ＿ｇｏｉｎｇ＿ｕｐ（）機能が呼び出されている。
【００２２】
さらに、ｒｏｕｔｅ＿ｔｏ＿ｃｎｓ（）機能に関していくつかの潜在的な欠点が生じる可能性がある。第一に、ｒｏｕｔｅ＿ｔｏ＿ｃｎｓ（）機能は記憶されておらず、エンドポートがどのエンドノードに属しているかを考慮することなくエンドポートをルーティングする。第二に、ｒｏｕｔｅ＿ｔｏ＿ｃｎｓ（）機能は、ルーティングについての物理的なポート番号に依存している。
【００２３】
図１は、ネットワーク環境においてファットツリー・ルーティングをサポートする例を示す。図１に示されるように、１つ以上のエンドノード１０１～１０４はネットワークファブリック１００に接続することができる。ネットワークファブリック１００は、ファットツリー・トポロジーに基づき得るものであって、複数のリーフスイッチ１１１～１１４および複数のスパインスイッチまたはルートスイッチ１３１～１３４を含む。加えて、ネットワークファブリック１００は、スイッチ１２１～１２４などの１つ以上の中間スイッチを含み得る。
【００２４】
また図１に示されるように、エンドノード１０１～１０４の各々は、マルチホーム型ノードであってもよく、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１００の２つ以上の部分に接続される単一ノードであってもよい。たとえば、ノード１０１は、ポートＨ１およびＨ２を含み得る。ノード１０２はポートＨ３およびＨ４を含み得る。ノード１０３はポートＨ５およびＨ６を含み得る。ノード１０４はポートＨ７およびＨ８を含み得る。
【００２５】
加えて、各々の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Ｓ１　１３１はスイッチポート１、２を有し得る。ルートスイッチＳ２　１３２はスイッチポート３、４を有し得る。ルートスイッチＳ３　１３３はスイッチポート５、６を有し得る。ルートスイッチＳ４　１３４はスイッチポート７、８を有し得る。
【００２６】
リーフスイッチベースでルーティングするアルゴリズム１などのファットツリー・ルーティングアルゴリズムを用いると、独立したルートが異なる２ポートノード１０１～１０４に割当てられることが保証されなくなる。たとえば、ポートＨ１、Ｈ２、Ｈ５およびＨ６は、各々のスイッチ（図示せず）上のポート１に接続することができ、ポートＨ３、Ｈ４、Ｈ７およびＨ８は、各々のスイッチ（図示せず）上のポート２に接続される。ここで、４つのエンドポートを介してルーティングし、リーフスイッチ１１３およびスイッチ１２３を通過した後、ファットツリー・ルーティングアルゴリズムは、ノード１０１上で対

をなすエンドポートＨ１およびＨ２からの２つの経路を、（それぞれ、スイッチポート１、２を介して）同じ最左端のルートスイッチにＳ１　１３１に割り当て得る。同様に、他のエンドポートの対、たとえば、ノード１０２上のＨ３およびＨ４、ノード１０３上のＨ５およびＨ６、ならびにノード１０４上のＨ７およびＨ８は、同じルートスイッチ（すなわち、Ｓ２　１３２～Ｓ４　１３４のそれぞれ）を介してルーティングされてもよい。

【００２７】

これにより、結果として、ユーザにとって不所望な動作が生じる可能性がある。図１に示されるように、エンドノード１０１は、当該エンドノード１０１が、固有の物理的なフォールトトレランス（すなわち、さまざまなリーフスイッチ１１１および１１３に接続された２つのエンドポート）を有する可能性があったとしても、ルートスイッチＳ１　１３１において単一障害点を被る可能性がある。加えて、物理的な配線に応じて、同様の問題が起こる可能性があり、単一障害点がファットツリー・トポロジーにおける他のスイッチ上で発生する可能性がある。

【００２８】

さらに、ネットワークファブリック１００内においては、同じノード上における異なるポートへのトラフィックが単一のリンクを介してルーティングされてもよい。このため、このような単一のリンクは、エンドポートの組についての付加的な単一障害点と、性能ボトルネックとの両方を呈する可能性がある（なぜなら、異なるエンドポートを対象としたトラフィックが有効に利用できるのが単一の共有リンクの帯域幅だけであるかもしれないからである）。

【００２９】

マルチホーム型ファットツリー・ルーティング

本発明の実施形態に従うと、システムは、ファットツリーにおけるマルチホーム型ノードのために独立したルートを提供することができ、そのため、単一障害点によって完全な機能停止がもたらされる可能性がなくなる。

【００３０】

図２は、本発明の実施形態に従った、ネットワーク環境におけるマルチホーム型ルーティングをサポートする例を示す。図２に示されるように、ネットワーク環境は、複数のエンドノード（たとえば、ノード２０１～２０４）を含み得る。これら複数のエンドノードは各々、１つ以上のポート（たとえば、ポートＨ１～Ｈ８）を含み得る。加えて、複数のエンドノード２０１～２０４は、ファットツリー・トポロジーにあり得るネットワークファブリック２００に接続することができる。また、ネットワークファブリック２００は、複数のスイッチ、たとえばリーフスイッチ２１１～２１４およびルートスイッチＳ１　２３１およびＳ２　２３２を含み得る。

【００３１】

本発明の実施形態に従うと、ｍＦｔｒｅｅアルゴリズムなどのマルチホーム型ファットツリー・ルーティングアルゴリズムは、たとえば、サブネットマネージャ（ＳＭ）２１０によって、ファットツリー・ルーティングを実行するために用いることができる。図２に示される例においては、ｍＦｔｒｅｅアルゴリズムは、相互に冗長なやり方でルーティングされる必要があり得るノード２０１上のポートＨ１およびポートＨ２からの経路を識別することができる。なぜなら、両方のポートが単一のエンドノード１上に位置しているからである。

【００３２】

さらに、ｍＦｔｒｅｅアルゴリズムは、経路を実際に冗長にすることを保証することができる。たとえば、ノード２０１上のポートＨ１からの経路は、リーフスイッチ２１１を通過し、最終的にルートスイッチ２３１に到達し得る。図２に示されるように、システムは、経路における（濃い陰影で示される）スイッチにマーク付けすることができる。そして、システムは、ノード２０１上のポートＨ２からの経路を決定するために、マーク付けされたスイッチを使用しないようにすることができる。これにより、ノード２０１上のポートＨ２からの経路は、（たとえば、リーフスイッチ２１３を介する）冗長経路を通過す

ることができ、最終的に、（軽く陰影を付けて示される）別のルートスイッチ２３２に到達する。

【００３３】

ノード２０１のためのルーティングステップが完了すると、アルゴリズムは、（太線で示されるように）ルーティングされたノードにマーク付けすることができる。これにより、アルゴリズムがノード２０１の別のポートに達したときに、このノード２０１についてルーティングステップが繰り返されないようにする。こうして、システムは、単一障害点が、マルチポートノードの完全な機能停止をもたらさないようにすることを保証することができる。

【００３４】

加えて、ファットツリー・ルーティングアルゴリズムは、インフィニバンド（ＩＢ）ファットツリー・トポロジーの性能、スケーラビリティ、利用可能性および予測可能性を改善させることができる。

【００３５】

以下のアルゴリズム２は、例示的なマルチホーム型ファットツリー・ルーティングアルゴリズムである。

【００３６】

【表２】

アルゴリズム２　*route_multihomed_cns() 機能*

要求: アドレス指定が完了

保証: すべての *hca_ports* が独立したスパインを介してルーティングされる

```
1: for swleaf = 0 to leaf_sw_num do
2:         for swleaf.port = 0 to max_ports do
3:                 hca_node = swleaf.port -> remote_node
4:                 if hca_node.routed ==  true then
5:                         continue
6:                 end if
7:                 route hcas(hca_node)
8:         end for
9: end for
```

【００３７】

上に示されるように、アルゴリズム２は、マルチホーム型ルーティングアルゴリズムであって、すべてのリーフスイッチに対して繰り返すことができ、さらに、各々のリーフスイッチのためのすべてのリーフスイッチポートに対して繰り返すことができる（１～９行目）。これにより、アルゴリズム２は、アルゴリズム１と同様に決定論的になり得る。

【００３８】

さらに、アルゴリズム２は、スイッチポートに関連付けられたエンドノードを発見するために、リーフスイッチ上のスイッチポートを採用することができる（３行目）。単にリーフスイッチに接続された遠隔ポートのＬＩＤを採用するアルゴリズム１とは異なり、アルゴリズム２は、ルーティング演算を実行するためのパラメータとしてエンドノードを採用することができる（７行目）。

【００３９】

以下のアルゴリズム３は、ファットツリーにおける単一のエンドノードをルーティングするための例示的なアルゴリズムである。

【００４０】

【表３】

アルゴリズム 3 *route_hcas(hca) 機能*

要求: ルーティングされるべきノード

保証: *hca_lid* を有するノードに属するすべての *hca_ports* がルーティングされる

```
1: for hca_node.port = 0 to port_num do
2:     hca_lid = hca_node.port -> lid
3:     swleaf = hca_node.port -> remote_node
4:     swleaf.port = hca_node.port -> remote_port_number
5:     swleaf.routing_table[hca_lid] = swleaf.port
6:     route_downgoing_by_going_up()
7: end for
8: hca_node.routed = true
9: clear_redundant_flag()
```

【００４１】

上に示されるように、アルゴリズム３は、選択されたエンドノード上のすべてのポートに対して繰り返すことができる（１～７行目）。たとえば、アルゴリズム３は、ｒｏｕｔｅ＿ｄｏｗｎｇｏｉｎｇ＿ｂｙ＿ｇｏｉｎｇ＿ｕｐ（）機能の修正バージョンを用いて、選択されたエンドノード上の各々のポートをルーティングすることができる（６行目）。選択されたエンドノード上のすべてのポートがルーティングされる場合、ルーティングアルゴリズムは、選択されたエンドノードをルーティングされるものとしてマーク付けすることができ（８行目）、これにより、別のリーフスイッチ上に到達したときにエンドノードがルーティングされないようにする。また、アルゴリズム３は、さまざまな状況でシステムの性能を改善させることができる（たとえば、アルゴリズム３は、２ポートノードのためにループ反復のうち半分を保存しておくことができる）。

【００４２】

加えて、アルゴリズム３は、単一のホストチャンネルアダプタ（ＨＣＡ：host channel adapter）上に複数のポートがあるシナリオと、２つ以上のＨＣＡ上に複数のポートがあるとシナリオとの両方に適用され得る。アルゴリズムはさまざまな方法を用いて、同じ論理ノード上の単一のＨＣＡまたは複数のＨＣＡ上のポート（またはいずれかのエンドポート）を識別することができる。ここで、ノードは、物理サーバもしくは仮想サーバ、ＩＯデバイス、または、１つ以上のＨＣＡポートを介してＩＢファブリックに接続された如何なるの種類のエンドノードであってもよい。

【００４３】

さらに、アルゴリズム３は、選択されたノード上の各々のポートをルーティングすることができ、フラグを用いて経路上の各々のスイッチにマーク付けすることができる。こうして、アルゴリズム３は、同じエンドノード上のさまざまなポートのためにさまざまなスイッチを選択することができる。その後、アルゴリズムは、次のノードにまで進み得るように、すべてのスイッチ上のフラグをひっくり返すことができる。

【００４４】

加えて、システムは、ｃｌｅａｒ＿ｒｅｄｕｎｄａｎｔ＿ｆｌａｇ（）機能においてスイッチ冗長性フラグをクリアすることによって最適化することができる（９行目）。スイッチ冗長性フラグをクリアするための最適な方法としては、特定のスイッチが経路上にあったか否かに関わらずすべてのスイッチに対して繰り返されるループをこの機能で用いるのではなく、経路上にあるスイッチのリストを作成して、ｃｌｅａｒ＿ｒｅｄｕｎｄａｎｔ＿ｆｌａｇ（）機能がリスト中のそれらスイッチに対してのみ繰り返されることを保証する方法が挙げられる。

【００４５】
以下のアルゴリズム４は、ファットツリーにおける単一のエンドノードポートをルーティングするための例示的なアルゴリズムである。
【００４６】
【表４】

アルゴリズム 4　*route_downgoing_by_going_up() 機能*

要求: 現在のホップスイッチ

保証: 上向きの冗長スイッチを発見するために最大限の努力が払われる

保証: 経路上のスイッチを冗長とマーク付けする

```
1: groupmin = 0
2: redundant_group = 0
3: for port_group = 0 to port_group_num do
4:         if groupmin == 0 then
5:                 if groupmin    -> remote_node.redundant then
6:                         groupmin = port_group
7:                 end if
8:         else if port_group.cntdown < groupmin.cntdown then
9:                 groupmin = port_group
10:                if groupmin -> remote_node.redundant then
11:                        min redundant group = groupmin
12:                end if
13:        end if
14: end for
15: if groupmin == 0 then
16:        fallback_normal_routing(hca_lid)
17: else if groupmin -> remote node.redundant then
18:        groupmin = min_redundant_group
19:        groupmin -> remote_node.redundant = false
20: end if
```

【００４７】
上に示されるように、アルゴリズム４におけるｒｏｕｔｅ＿ｄｏｗｎｇｏｉｎｇ＿ｂｙ＿ｇｏｉｎｇ＿ｕｐ（）機能の修正バージョンでは、冗長性が主要な問題として処理される。最低位の下方向カウンタを有するポートグループが選択されているアルゴリズム１とは異なり、アルゴリズム４は、当該アルゴリズム４がエンドノードに属する他のいずれのポートもルーティングしない場合（すなわち、冗長フラグが真である場合）、エンドノードの上向きノードを次のホップとして選択し得るだけである。
【００４８】
ここで、次のエンドノードがルーティングされる前に、冗長フラグがクリアされる。極めて過度にサブスクライブされたファブリックにおいて起こり得るような冗長なノードが存在しない場合、または、リンク障害がある場合、ｍＦｔｒｅｅは通常のファットツリー・ルーティングにまで後退し、この場合、ユーザは、アルゴリズム１において提示されるルーティング機能と同様にこのルーティング機能が動作することを認識し得る。
【００４９】

加えて、２つのスイッチ間に代替的なスイッチおよび平行なリンクが存在しない場合、上述のアルゴリズム４は、性能／負荷の分散およびリンクレベル冗長性の両方をサポートするために、同じエンドノード上のさまざまな目標ポートのためにさまざまなリンクを選択することができる。他方では、性能が優先される場合、両方のリンクが同じレベルの負荷を有する場合にのみ、同じエンドノード上の異なる目標ポートに対して別個のリンクが選択され得る。

【００５０】

図３は、ネットワーク環境においてファットツリー・ルーティングをサポートするために冗長性を提供する例を示す。図３に示されるように、１つ以上のマルチホーム型エンドノード３０１～３０４はネットワークファブリック３００に接続することができる。ネットワークファブリック３００は、ファットツリー・トポロジーに基づくものであって、複数のリーフスイッチ３１１～３１２と、複数のスパインスイッチまたはルートスイッチＳ１　３３１～Ｓ４　３３４とを含み得る。加えて、ネットワークファブリック３００は、スイッチ３２１～３２４などの１つ以上の中間スイッチを含み得る。

【００５１】

図３にも示されるように、ノード３０１はポートＨ１およびＨ２を含み得る。ノード３０２はポートＨ３およびＨ４を含み得る。ノード３０３はポートＨ５およびＨ６を含み得る。また、ノード３０４はポートＨ７およびＨ８を含み得る。加えて、ルートスイッチＳ１　３３１はスイッチポート１、２を有し得る。ルートスイッチＳ２　３３２はスイッチポート３、４を有し得る。ルートスイッチＳ３　３３３はスイッチポート５、６を有し得る。ルートスイッチＳ４　３３４はスイッチポート７、８を有し得る。

【００５２】

さらに、ｍＦｔｒｅｅアルゴリズムなどのマルチホーム型ルーティングは、ノード上の各々のポートへの経路が排他的となるように、各々のマルチホーム型ノード３０１～３０４をルーティングすることができる。すなわち、ｍＦｔｒｅｅアルゴリズムは、独立した経路を介してマルチホーム型ノード上の各ポートに到達可能となるように保証する。加えて、ｍＦｔｒｅｅアルゴリズムはネットワーク性能を改善させることができる。

【００５３】

さらに、単一のマルチホーム型エンドノードの場合、ｍＦｔｒｅｅアルゴリズムは、同じエンドノードに属するいずれのポート対に至る経路によっても単一のリンクが共有されないように保証することができる。また、ネットワークファブリック３００において同じ宛先ノード上に異なるソースポートから異なるポートに至るコンカレントなトラフィックがある場合、ｍＦｔｒｅｅアルゴリズムは、代替的な経路が存在する場合に、コンカレントなトラフィックが如何なる中間リンクも共有しないことを保証することができる。

【００５４】

このように、ｍＦｔｒｅｅアルゴリズムを用いると、ファブリック３００におけるスパインスイッチＳ１　３３１などの単一デバイスが故障しても、ノード３０１の切断が引き起こされる可能性はない。なぜなら、異なるポートへの経路が単一のスパインスイッチＳ１　３３１においては収束しないからである。

【００５５】

加えて、ｍＦｔｒｅｅアルゴリズムは、別個の独立したエンティティと同じノード上の各々のポートを処理する。これにより、ｍＦｔｒｅｅアルゴリズムは、ポートベースではなくノードベースでルーティングすることができ、ｍＦｔｒｅｅアルゴリズムは、異なるエンドノードが有し得るさまざまな特徴をアドレス指定することができる。

【００５６】

図４は、本発明の実施形態に従った、ネットワーク環境においてマルチホーム型ルーティングをサポートするための例示的なフローチャートを示す。図４に示されるように、ステップ４０１において、システムは、ネットワークファブリックにおいてリーフスイッチ上のスイッチポートに関連付けられたエンドノードを提供することができる。この場合、エンドノードは複数のポートに関連付けられている。次いで、ステップ４０２において、

システムは、エンドノード上の各々の上記ポートのためにルーティングを実行することができる。さらに、ステップ４０３において、システムは、エンドノード上の複数のポートが相互に独立した経路を取ることを保証することができる。

【００５７】

図５は、本発明の実施形態が実現され得るコンピュータシステムのブロック図を示す。コンピュータシステム５００は、情報を通信するためのバスまたは他の通信メカニズムと、バスと結合されて情報を処理するためのハードウェアプロセッサとを含む。ハードウェアプロセッサは、たとえば、汎用のマイクロプロセッサであってもよい。

【００５８】

コンピュータシステム５００はまた、プロセッサによって実行されるべき命令および情報を格納するための、バスに結合されたランダムアクセスメモリ（ＲＡＭ：random access memory）または他の動的記憶装置などのメインメモリを含む。メインメモリはまた、プロセッサによって実行される命令の実行中に一時変数または他の中間情報を格納するために用いられてもよい。このような命令がプロセッサにアクセス可能な非一時的な記憶媒体に格納されると、コンピュータシステム５００が、命令で指定された動作を実行するようにカスタマイズされた専用マシンとなる。

【００５９】

コンピュータシステム５００はさらに、プロセッサのために静的情報および命令を格納するための、バスに結合された読取専用メモリ（ＲＯＭ：read only memory）または他の静的記憶装置を含む。情報および命令を格納するための、バスに結合された磁気ディスクまたは光ディスクなどの記憶装置が提供される。

【００６０】

コンピュータシステム５００は、情報をコンピュータユーザに対して表示するための、陰極線管（ＣＲＴ：cathode ray tube）などのディスプレイを含み得るか、または、当該ディスプレイに対してバスを介して結合され得る。英数字および他のキーもしくはカーソル制御を含む入力装置は、プロセッサに対して情報およびコマンド選択を伝えるためにバスに結合される。

【００６１】

コンピュータシステム５００はこの明細書中に記載される技術を実現し得るが、この場合、コンピュータシステムと組み合わせることで、コンピュータシステム５００を専用マシンにするかまたは専用マシンにするようプログラムする、カスタマイズされたハードワイヤード論理、１つ以上のＡＳＩＣもしくはＦＰＧＡ、ファームウェアおよび／またはプログラム論理を用い得る。一実施形態に従うと、この明細書中に記載される技術は、プロセッサがメインメモリに含まれる１つ以上の命令の１つ以上のシーケンスを実行することに応じて、コンピュータシステム５００によって実行される。このような命令は、記憶装置などの別の記憶媒体からメインメモリに読込まれてもよい。メインメモリに含まれる命令のシーケンスを実行することにより、この明細書中に記載されるプロセスステップをプロセッサに実行させる。代替的な実施形態においては、ハードワイヤード回路が、ソフトウェア命令の代わりに、またはソフトウェア命令と組み合わせて用いられてもよい。

【００６２】

コンピュータシステム５００はまた、バスに連結された通信インターフェイスを含む。通信インターフェイスは、統合サービスデジタル網（ＩＳＤＮ：integrated services digital network）カード、ケーブルモデム、衛星モデム、または対応するタイプの電話線にデータ通信接続するためのモデムであってもよい。別の例として、通信インターフェイスは、互換性のあるローカルエリアネットワーク（ＬＡＮ：local area network）にデータ通信接続するためのＬＡＮカードであってもよい。ワイヤレスリンクが実現されてもよい。このような如何なる実現例においても、通信インターフェイスは、さまざまなタイプの情報を表わすデジタルデータストリームを搬送する電気信号、電磁信号または光信号を送受信する。

【００６３】

図５に示されるように、記憶装置は、ネットワークファブリックにおいてリーフスイッチ上のスイッチポートにエンドノードを関連付けるように構成され得るサブネットマネージャを含んでもよい。この場合、エンドノードは、複数のポートに関連付けられている。サブネットマネージャはさらに、エンドノード上の各々の上記ポートのためにルーティングを実行し、エンドノード上の複数のポートが相互に独立した経路を取ることを保証するように構成され得る。

【００６４】

いくつかの実施形態においては、サブネットマネージャはさらに、エンドノード上の複数のポートにおける或るポートに関連付けられた経路上の各々のスイッチにマーク付けし、エンドノード上の複数のポートにおける別のポートに関連付けられた別の経路が、マーク付けされたスイッチを用いることを防止するように構成されてもよい。

【００６５】

いくつかの実施形態においては、サブネットマネージャはさらに、冗長なスイッチがない場合に、エンドノード上の複数のポートにおける別のポートに関連付けられた別の経路が、１つ以上のマーク付けされたスイッチを用いることを可能にし、平行なリンクが１つ以上のマーク付けされたスイッチ上に存在する場合に、同じエンドノード上の異なる目標ポートのために独立したリンクを選択するように構成されてもよい。

【００６６】

いくつかの実施形態においては、サブネットマネージャはさらに、エンドノード上の複数のポートのルーティングを完了した後、各々のマーク付けされたスイッチのマーク付けを解除するように構成されてもよい。

【００６７】

いくつかの実施形態においては、サブネットマネージャはさらに、エンドノード上の複数のポートのルーティングを完了した後、エンドノードをルーティングされたエンドノードとしてマーク付けするように構成されてもよい。

【００６８】

いくつかの実施形態においては、サブネットマネージャはさらに、エンドノードが別のリーフスイッチに遭遇したときに、エンドノードが再びルーティングされることを防止するように構成されてもよい。

【００６９】

図６は、本発明の実施形態を実現するためのシステム６００のブロック図を示す。システム６００のブロックは、本発明の原理を実行するために、ハードウェア、ソフトウェア、またはハードウェアとソフトウェアとの組み合わせによって実現されてもよい。上述のとおり本発明の原理を実現するために、図６において記載されたブロックが組み合わされ得るかまたはサブブロックに分解され得ることが当業者によって理解される。したがって、この明細書中における説明は、この明細書中に記載される機能ブロックの実現可能な任意の組み合わせまたは分解またはさらなる定義をサポートし得る。

【００７０】

図６に示されるように、システム６００は、ネットワークファブリックにおいてリーフスイッチ上のスイッチポートに関連付けられたエンドノードを提供するように構成された提供ユニット６０１を含み得る。エンドノードは、複数のポートに関連付けられている。システム６００はさらに、エンドノード上の各々の上記ポートのためにルーティングを実行するように構成された実行ユニット６０２を含み得る。さらに、システム６００は、エンドノード上の複数のポートが相互に独立した経路を取ることを保証するように構成された保証ユニット６０３をさらに含み得る。

【００７１】

図６には示されないが、いくつかの実施形態においては、システム６００はさらに、ネットワークファブリックをファットツリー・トポロジーに基づいたものにすることを可能にするように構成された第１の可能ユニットを含み得る。

【００７２】

いくつかの実施形態においては、システム６００はさらに、エンドノードが、複数のポートを介してファットツリー・トポロジーの２つ以上の部分に接続されるマルチホーム型ノードとなることを可能にするように構成された第２の可能ユニットを含み得る。

【００７３】

いくつかの実施形態においては、システム６００はさらに、エンドノード上の複数のポートにおける或るポートに関連付けられた経路上の各々のスイッチにマーク付けするように構成された第１のマーク付けユニットと、エンドノード上の複数のポートにおける別のポートに関連付けられた別の経路が当該マーク付けされたスイッチを用いることを防止するように構成された第１の防止ユニットとを含み得る。

【００７４】

いくつかの実施形態においては、システム６００はさらに、冗長なスイッチがない場合に、エンドノード上の複数のポートにおける別のポートに関連付けられた別の経路が１つ以上のマーク付けされたスイッチを用いることを可能にするように構成された第３の可能ユニットと、平行なリンクが１つ以上のマーク付けされたスイッチ上に存在する場合に、同じエンドノード上の異なる目標ポートのために独立したリンクを選択するように構成された選択ユニットとを含み得る。

【００７５】

いくつかの実施形態においては、システム６００はさらに、エンドノード上の複数のポートのルーティングを完了した後、各々のマーク付けされたスイッチのマーク付けを解除するように構成されたマーク付け解除ユニットを含み得る。

【００７６】

いくつかの実施形態においては、システム６００はさらに、エンドノードからの相互に独立した各々の経路を異なるスパインスイッチに関連付けるように構成された関連付けユニットを含み得る。

【００７７】

いくつかの実施形態においては、システム６００はさらに、エンドノード上の複数のポートのルーティングを完了した後、エンドノードをルーティングされたエンドノードとしてマーク付けするように構成された第２のマーク付けユニットを含み得る。

【００７８】

いくつかの実施形態においては、システム６００はさらに、エンドノードが別のリーフスイッチに遭遇したときに、エンドノードが再びルーティングされることを防止するように構成された第２の防止ユニットを含み得る。

【００７９】

いくつかの実施形態においては、システム６００はさらに、ルーティングアルゴリズムがエンドノードをパラメータとして採用することを可能にするように構成された第４の可能ユニットを含み得る。

【００８０】

ネットワーク環境においてマルチホーム型ルーティングをサポートするための手段は、ネットワークファブリックにおいてリーフスイッチ上のスイッチポートに関連付けられたエンドノードを提供することを含む。エンドノードは複数のポートに関連付けられている。当該手段はさらに、エンドノード上の各々の上記ポートのためにルーティングを実行することと、エンドノード上の複数のポートが上記ルーティングのために相互に独立した経路を取ることを保証することとを含む。

【００８１】

ネットワーク環境においてマルチホーム型ルーティングをサポートするための手段においては、ネットワークファブリックはファットツリー・トポロジーに基づいている。

【００８２】

ネットワーク環境においてマルチホーム型ルーティングをサポートするための手段においては、エンドノードは、複数のポートを介してファットツリー・トポロジーの２つ以上の部分に接続されたマルチホーム型ノードである。

【００８３】
ネットワーク環境においてマルチホーム型ルーティングをサポートするための手段はさらに、エンドノード上の複数のポートにおける或るポートに関連付けられた経路上の各々のスイッチにマーク付けすることと、エンドノード上の複数のポートにおける別のポートに関連付けられた別の経路が、マーク付けされたスイッチを用いることを防止することとを含む。
【００８４】
ネットワーク環境においてマルチホーム型ルーティングをサポートするための手段はさらに、冗長なスイッチがない場合に、エンドノード上の複数のポートにおける別のポートに関連付けられた別の経路が、１つ以上のマーク付けされたスイッチを用いることを可能にすることと、平行なリンクが１つ以上のマーク付けされたスイッチ上に存在する場合に、同じエンドノード上の異なる目標ポートのために独立したリンクを選択することとを含む。
【００８５】
ネットワーク環境においてマルチホーム型ルーティングをサポートするための手段はさらに、エンドノード上の複数のポートのルーティングを完了した後に各々のマーク付けされたスイッチのマーク付けを解除することをさらに含む。
【００８６】
ネットワーク環境においてマルチホーム型ルーティングをサポートするための手段はさらに、エンドノードからの相互に独立した経路を異なるスパインスイッチに関連付けることを含む。
【００８７】
ネットワーク環境においてマルチホーム型ルーティングをサポートするための手段はさらに、エンドノード上の複数のポートのルーティングを完了した後、エンドノードをルーティングされたエンドノードとしてマーク付けすることを含む。
【００８８】
ネットワーク環境においてマルチホーム型ルーティングをサポートするための手段はさらに、エンドノードが別のリーフスイッチに遭遇した場合に、エンドノードが再びルーティングされることを防止することを含む。
【００８９】
ネットワーク環境においてマルチホーム型ルーティングをサポートするための手段はさらに、エンドノードをパラメータとして採用するルーティングアルゴリズムを用いることを含む。
【００９０】
本発明は、本開示の教示に従いプログラムされた、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可能な記録媒体を含む、従来の汎用もしくは専用デジタルコンピュータ、コンピューティングデバイス、マシン、またはマイクロプロセッサを１つ以上用いて、適宜実装し得る。適切なソフトウェアコーディングは、熟練したプログラマが本開示の教示に基づいて容易に準備できるものである。これはソフトウェア技術における当業者には明らかであろう。
【００９１】
実施形態によっては、本発明は、本発明のプロセスのうちいずれかを実行するためにコンピュータをプログラムするのに使用できる命令が格納された記録媒体または（１つまたは複数の）コンピュータ読取可能な媒体であるコンピュータプログラムプロダクトを含む。この記録媒体の例は、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、任意の種類のディスク、ＲＯＭ、ＲＡＭ、ＥＰＲＯＭ、ＥＥＰＲＯＭ、ＤＲＡＭ、ＶＲＡＭ、フラッシュメモリデバイス、磁気もしくは光カード、ナノシステム（分子メモリＩＣを含む）、または、命令および／またはデータを格納するのに適した任意の種類の媒体もしくはデバイスを含み得るものの、これらに限定されない。

【００９２】

本発明のこれまでの記載は例示および説明を目的として提供されている。すべてを網羅するまたは本発明を開示された形態そのものに限定することは意図されていない。当業者には数多くの変更および変形が明らかであろう。実施の形態は、本発明の原理およびその実際の応用を最もうまく説明することによって他の当業者がさまざまな実施の形態および意図している特定の用途に適したさまざまな変形を理解できるようにするために、選択され説明されている。

【図１】

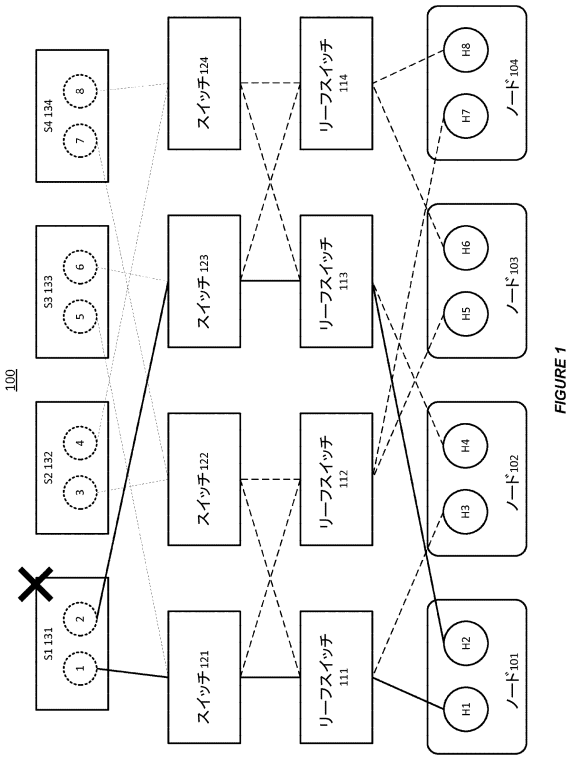

FIGURE 1

【図２】

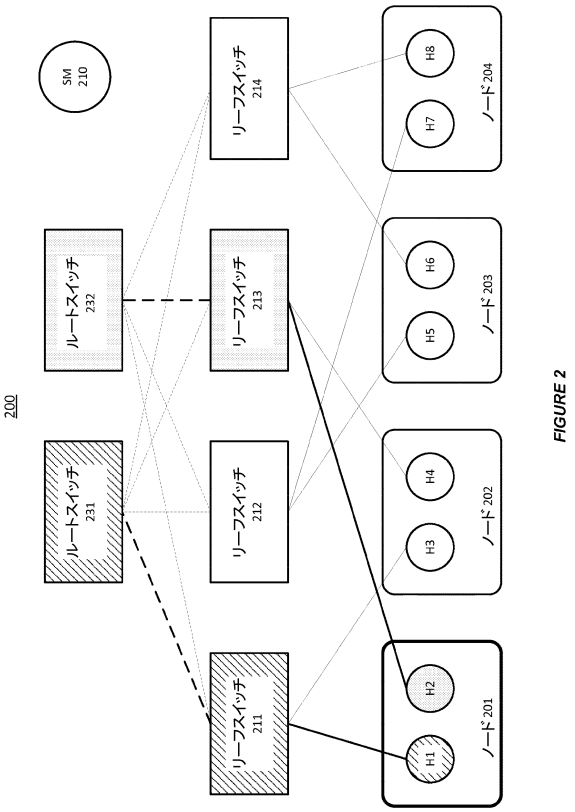

FIGURE 2

【図３】

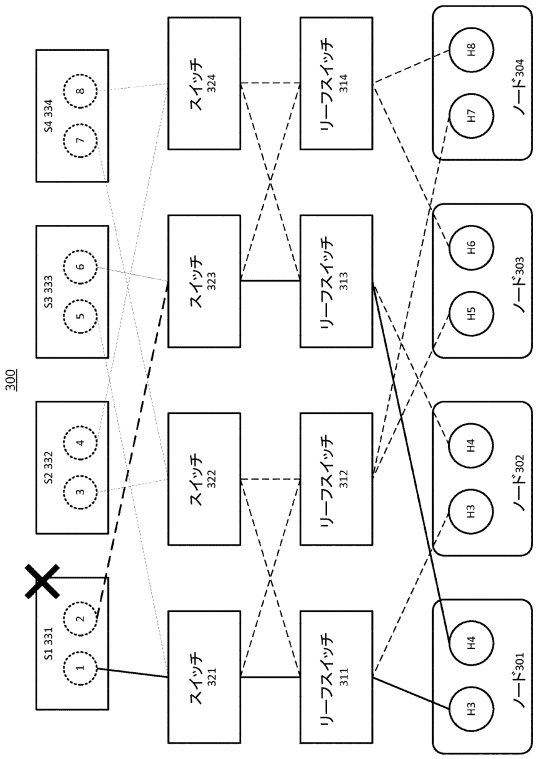

FIGURE 3

【図４】

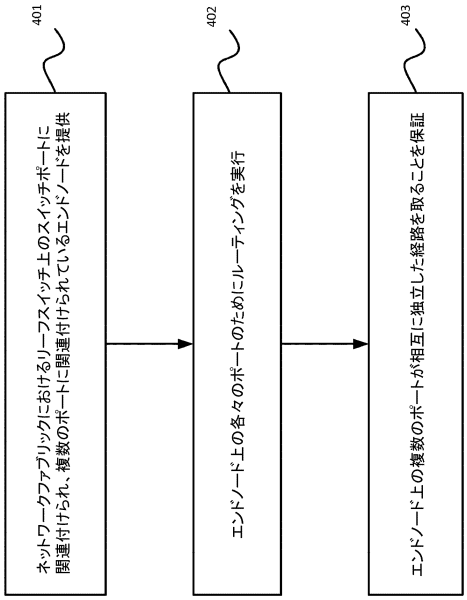

FIGURE 4

【図５】

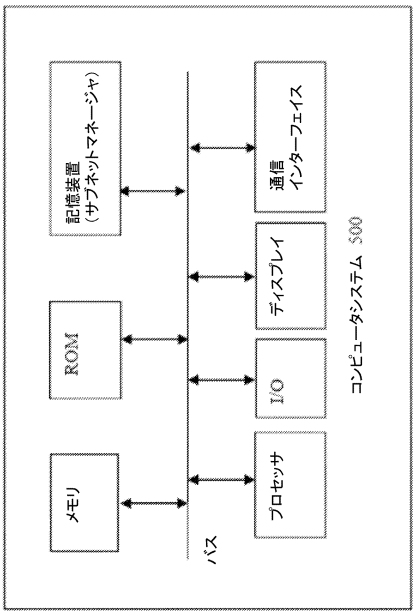

FIGURE 5

【図６】

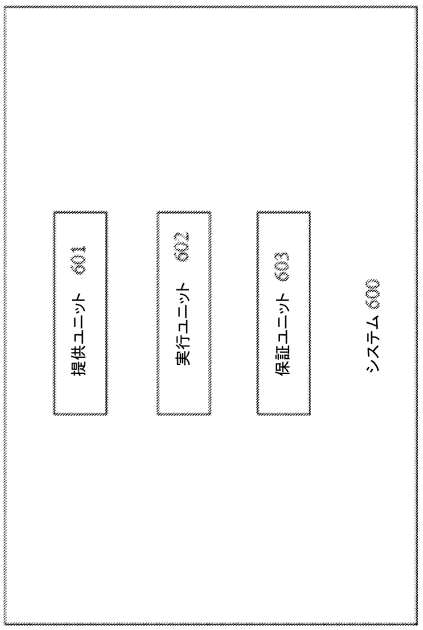

FIGURE 6

フロントページの続き

審査官　野元　久道

(56)参考文献　米国特許出願公開第２０１２／０３００６６９（ＵＳ，Ａ１）
特開２００７－３１８４４９（ＪＰ，Ａ）
特開２０１３－０５１６４７（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　１２／４４