【公報種別】特許法第１７条の２の規定による補正の掲載
【部門区分】第７部門第３区分
【発行日】平成28年5月26日(2016.5.26)

【公表番号】特表2016-510549(P2016-510549A)
【公表日】平成28年4月7日(2016.4.7)
【年通号数】公開・登録公報2016-021
【出願番号】特願2015-553847(P2015-553847)
【国際特許分類】
Ｈ０４Ｌ　12/751　　(2013.01)
【ＦＩ】
Ｈ０４Ｌ　12/751

【誤訳訂正書】
【提出日】平成28年3月22日(2016.3.22)
【誤訳訂正１】
【訂正対象書類名】特許請求の範囲
【訂正対象項目名】全文
【訂正方法】変更
【訂正の内容】
【特許請求の範囲】
【請求項１】
ネットワークオンチップ（ＮｏＣ）トポロジのホスト位置を決定するための方法であって、
プロセッサを利用して、
前記ＮｏＣトポロジの複数のホストから、割り当てられた重みに基づいて第１のホストを選択し、
前記ＮｏＣトポロジの前記複数のホストから、第２のホストに隣接する複数の位置から１の隣接位置を選択し、
前記第２のホストは、前記複数のホストのうちの少なくとも一つに割り当てられた第１の確率受付関数からの少なくとも一つの確率から決定されており、前記隣接位置の選択は、前記第２のホストに隣接する前記複数の位置の少なくとも一つに割り当てられた第２の確率受付関数からの少なくとも一つの確率に基づいており、
前記第１のホストの移動を受け入れることを示す結果である再配置決定を行う第３の確率受付関数に基づいた再配置決定のために前記隣接位置に前記第１のホストを移動する、ＮｏＣトポロジのホスト位置を決定するための方法。
【請求項２】
前記方法は、
反復の最大反復回数に到達するまで、ホストの位置を決定するための前記方法を反復的に繰り返し、
ホスト間の通信における、通信帯域幅、前記ＮｏＣトポロジおよび１以上のレイテンシ制約の少なくとも一つに基づいて、前記第１の確率受付関数と前記第２の確率受付関数とを更新すること、
をさらに含む、
請求項１に記載の方法。
【請求項３】
前記方法は、
前記ＮｏＣトポロジの演算に用いる温度パラメータが最低温度閾値に達するまでホスト位置を決定するための前記方法を反復的に繰り返し、
前記ＮｏＣトポロジの前記温度パラメータ、ホスト間の通信における、通信帯域幅、前

記ＮｏＣトポロジおよび１以上のレイテンシ制約の少なくとも一つに基づいて、前記第１の確率受付関数と前記第２の確率受付関数を更新すること、
をさらに含む、
請求項１に記載の方法。
【請求項４】
前記第１の確率受付関数は、
前記第１のホストが前記複数のホストのうちの少なくとも１つと通信する頻度、前記第１のホストと前記複数のホストのうちの少なくとも１つとの間の通信におけるレイテンシ制約、および前記第１のホストと前記複数のホストのうちの少なくとも１つとの間の帯域幅の制約の少なくとも一つに基づいて、前記複数のホストのうち少なくとも一つに少なくとも一つの確率を割り当てる、
請求項１に記載の方法。
【請求項５】
前記第２の確率受付関数は、
前記第２のホストが前記第２のホストと隣接する前記複数の位置のうちの少なくとも一つと通信する頻度、前記第２のホストと前記第２のホストに隣接する前記複数の位置のうちの前記少なくとも一つとの間の通信におけるレイテンシ制約、および前記第２のホストと前記第２のホストに隣接する前記複数の位置のうちの前記少なくとも一つとの間の帯域幅の制約の少なくとも一つに基づいて、前記第２のホストに隣接する複数の位置の前記少なくとも一つに前記少なくとも一つの確率を割り当てる、
請求項１に記載の方法。
【請求項６】
前記第１の確率受付関数、前記第２の確率受付関数、および前記第３の確率受付関数は、遺伝的最適化アルゴリズムと機械学習の少なくとも一つに基づいて実装されている、
請求項１に記載の方法。
【請求項７】
前記第３の確率受付関数は、コスト関数の減少と前記ＮｏＣトポロジの演算に用いる温度パラメータとの少なくとも一つに基づいて再配置の確率を提供する、
請求項１に記載の方法。
【請求項８】
ネットワークオンチップ（ＮｏＣ）トポロジのホスト位置をコンピュータに決定させるための指示を記憶する非一時的なコンピュータ可読記憶媒体であって、
前記指示は、
前記ＮｏＣトポロジの複数のホストから、割り当てられた重みに基づいて第１のホストを選択し、
前記ＮｏＣトポロジの前記複数のホストから、第２のホストに隣接する複数の位置から１の隣接位置を選択し、
前記第２のホストは、前記複数のホストのうちの少なくとも一つに割り当てられた第１の確率受付関数からの少なくとも一つの確率から決定されており、前記隣接位置の選択は、前記第２のホストに隣接する前記複数の位置の少なくとも一つに割り当てられた第２の確率受付関数からの少なくとも一つの確率に基づいており、
前記第１のホストの移動を受け入れることを示す結果である再配置決定を行う第３の確率受付関数に基づいた再配置決定のために前記隣接位置に前記第１のホストを移動すること、を含む、
非一時的なコンピュータ可読記憶媒体。
【請求項９】
前記指示は、
反復の最大反復回数に到達するまで、ホストの位置を決定するための前記指示を反復的に繰り返し、
ホスト間の通信における、通信帯域幅、前記ＮｏＣトポロジおよび１以上のレイテンシ

制約の少なくとも一つに基づいて、前記第１の確率受付関数と前記第２の確率受付関数とを更新すること、
をさらに含む、
請求項８に記載の非一時的なコンピュータ可読記憶媒体。
【請求項１０】
前記指示は、
前記ＮｏＣトポロジの演算に用いる温度パラメータが最低温度閾値に達するまでホスト位置を決定するための前記指示を反復的に繰り返し、
前記ＮｏＣトポロジの前記温度パラメータ、ホスト間の通信における、通信帯域幅、前記ＮｏＣトポロジおよび１以上のレイテンシ制約の少なくとも一つに基づいて、前記第１の確率受付関数と前記第２の確率受付関数を更新すること、
をさらに含む、
請求項８に記載の非一時的なコンピュータ可読記憶媒体。
【請求項１１】
前記第１の確率受付関数は、
前記第１のホストが前記複数のホストのうちの少なくとも１つと通信する頻度、前記第１のホストと前記複数のホストのうちの少なくとも１つとの間の通信におけるレイテンシ制約、および前記第１のホストと前記複数のホストのうちの少なくとも１つとの間の帯域幅の制約の少なくとも一つに基づいて、前記複数のホストのうち少なくとも一つに少なくとも一つの確率を割り当てる、
請求項８に記載の非一時的なコンピュータ可読記憶媒体。
【請求項１２】
前記第２の確率受付関数は、
前記第２のホストが前記第２のホストと隣接する前記複数の位置のうちの少なくとも一つと通信する頻度、前記第２のホストと前記第２のホストに隣接する前記複数の位置のうちの前記少なくとも一つとの間の通信におけるレイテンシ制約、および前記第２のホストと前記第２のホストに隣接する前記複数の位置のうちの前記少なくとも一つとの間の帯域幅の制約の少なくとも一つに基づいて、前記第２のホストに隣接する複数の位置の前記少なくとも一つに前記少なくとも一つの確率を割り当てる、
請求項８に記載の非一時的なコンピュータ可読記憶媒体。
【請求項１３】
前記第１の確率受付関数、前記第２の確率受付関数、および前記第３の確率受付関数は、遺伝的最適化アルゴリズムと機械学習の少なくとも一つに基づいて実装されている、
請求項８に記載の非一時的なコンピュータ可読記憶媒体。
【請求項１４】
前記第３の確率受付関数は、コスト関数の減少と前記ＮｏＣトポロジの演算に用いる温度パラメータとの少なくとも一つに基づいて再配置の確率を提供する、
請求項８に記載の非一時的なコンピュータ可読記憶媒体。
【請求項１５】
ネットワークオンチップ（ＮｏＣ）トポロジのホスト位置を決定するための再配置ホスト選択モジュールを含むシステムであって、
前記再配置ホスト選択モジュールは、
前記ＮｏＣトポロジの複数のホストから、割り当てられた重みに基づいて第１のホストを選択し、
前記ＮｏＣトポロジの前記複数のホストから、第２のホストに隣接する複数の位置から１の隣接位置を選択し、
前記第２のホストは、前記複数のホストのうちの少なくとも一つに割り当てられた第１の確率受付関数からの少なくとも一つの確率から決定されており、前記隣接位置の選択は、前記第２のホストに隣接する前記複数の位置の少なくとも一つに割り当てられた第２の確率受付関数からの少なくとも一つの確率に基づいており、

再配置受付関数モジュールは、前記第１のホストの移動を受け入れることを示す結果である再配置決定を行う第３の確率受付関数に基づいて前記第１のホストを一つの位置から他の位置に移動するよう構成される、
システム。
【請求項１６】
前記再配置ホスト選択モジュールはさらに、
反復の最大反復回数に到達するまでホストの位置を反復的に決定し、
ホスト間の通信における、通信帯域幅、前記ＮｏＣトポロジおよび１以上のレイテンシ制約の少なくとも一つに基づいて、前記第１の確率受付関数と前記第２の確率受付関数とを更新するように構成される、
請求項１５に記載のシステム。
【請求項１７】
前記再配置ホスト選択モジュールはさらに、
前記ＮｏＣトポロジの演算に用いる温度パラメータが最低温度閾値に達するまでホスト位置を反復的に決定し、
前記ＮｏＣトポロジの前記温度パラメータ、ホスト間の通信帯域幅、前記ＮｏＣトポロジおよび１以上の通信におけるレイテンシ制約の少なくとも一つに基づいて、前記第１の確率受付関数と前記第２の確率受付関数を更新するように構成される、
請求項１５に記載のシステム。
【請求項１８】
前記第１の確率受付関数は、
前記第１のホストが前記複数のホストのうちの少なくとも１つと通信する頻度、前記第１のホストと前記複数のホストのうちの少なくとも１つとの間の通信におけるレイテンシ制約、および前記第１のホストと前記複数のホストのうちの少なくとも１つとの間の帯域幅の制約の少なくとも一つに基づいて、前記複数のホストのうち少なくとも一つに少なくとも一つの確率を割り当てる、
請求項１５に記載のシステム。
【請求項１９】
前記第２の確率受付関数は、
前記第２のホストが前記第２のホストと隣接する前記複数の位置のうちの少なくとも一つと通信する頻度、前記第２のホストと前記第２のホストに隣接する前記複数の位置のうちの前記少なくとも一つとの間の通信におけるレイテンシ制約、および前記第２のホストと前記第２のホストに隣接する前記複数の位置のうちの前記少なくとも一つとの間の帯域幅の制約の少なくとも一つに基づいて、前記第２のホストに隣接する複数の位置の前記少なくとも一つに前記少なくとも一つの確率を割り当てる、
請求項１５に記載のシステム。
【請求項２０】
前記第１の確率受付関数、前記第２の確率受付関数、および前記第３の確率受付関数は、遺伝的最適化アルゴリズムと機械学習の少なくとも一つに基づいて実装されている、
請求項１５に記載のシステム。
【請求項２１】
前記第３の確率受付関数は、コスト関数の減少と前記ＮｏＣトポロジの演算に用いる温度パラメータとの少なくとも一つに基づいて再配置の確率を提供する、
請求項１５に記載のシステム。
【誤訳訂正２】
【訂正対象書類名】明細書
【訂正対象項目名】全文
【訂正方法】変更
【訂正の内容】
【発明の詳細な説明】

【発明の名称】全体的なレイテンシを最小限に抑え、相互接続コストを削減するために、自動接続ＳＯＣＳ　ＩＰコアをノードに相互接続する方法
【技術分野】
【０００１】
本明細書に記載された方法及び実装例は、アーキテクチャを相互接続するように指示されており、より具体的には、ネットワークオンチップＮｏＣシステムの相互接続アーキテクチャにおいて、システム・オン・チップ知的財産（ＳｏＣ　ＩＰ）コアをネットワークオンチップ（ＮｏＣ）ノードへ自動的に接続するためのものである。
【背景技術】
【０００２】
複雑化したシステムと縮小したトランジスタジオメトリによる統合レベルの増大によって、チップ上のコンポーネント数は急速に成長している。複雑なシステムオンチップ（ＳｏＣｓ）は、例えば、プロセッサコア、ＤＳＰｓ、ハードウェア・アクセラレータ、メモリそしてＩ／Ｏ等の様々なコンポーネントを含んでもよい。一方、チップマルチプロセッサ（ＣＭＰｓ）は、多数の同種のプロセッサコア、メモリそしてＩ／Ｏサブシステムを含んでもよい。
【０００３】
ＳｏＣとＣＭＰシステムの両方で、様々なコンポーネント間において高性能な通信を提供することに、オンチップ相互接続が役割を果たしている。従来のバスのスケーラビリティの制限とクロスバー基準の相互接続のために、チップ上の多数の部品を相互接続するための典型として、ネットワークオンチップ（ＮｏＣ）が登場した。ＮｏＣは、ポイントツーポイントの物理リンクを使用して互いに相互接続された、いくつかのルーティングノードで構成される、グローバルな共有通信インフラストラクチャである。メッセージは送信元によって注入され、複数の中間ノードとの物理リンク上の宛先へソースノードからルーティングされる。
【０００４】
宛先ノードは、メッセージを出し、宛先にメッセージを提供する。本出願の残りの部分については、用語「コンポーネント」、「ブロック」、「ホスト」または「コア」がＮｏＣを使用して相互接続されている様々なシステム構成要素を交互に参照して使用される。用語「ルータ」と「ノード」はまた、相互に使用される。一般化を失うことなく、複数の相互接続コンポーネントを備えたシステムは、それ自体を「マルチコアシステム」と呼ぶ。
【０００５】
システムネットワークを作成するために、ルータが相互に接続することができる、いくつかのトポロジがある。双方向リング（図１（ａ）参照）、２－Ｄ（二次元）メッシュ（図１（ｂ）参照）は、従来のトポロジの２次元トーラス（図１（ｃ）参照）の例である。メッシュとトーラスはまた、２．５Ｄ（２．５次元）または３Ｄ（三次元）構成に拡張される。図１（ｄ）は、３層の３×３の２ＤメッシュＮｏＣがそれぞれの上に示された３ＤメッシュＮｏＣを示している。ＮｏＣルータは、最大２つの追加ポートを持っている。１個のポートは上層でルータに接続し、別のポートは下層でルータに接続する。例えば、中間層におけるルータ１１１は、両方が使用されたポートを持っている。１個のポートは上層でルータに接続し、別のポートは下層でルータに接続する。ルータ１１０及び１１２は、それぞれ下層および上層メッシュ層である。したがって、それらはそれぞれ接続されている、上層に面したポート１１３と下層に面したポート１１４とを唯一持っている。
【０００６】
パケットは、様々なコンポーネント間の相互通信のためのメッセージ転送単位である。ルーティングは、ルータやパケットが送信元から宛先に送信されるネットワークの物理的なリンクの集合で構成されるパスを識別することを含む。コンポーネントは、１つまたは複数のルータの１つまたは複数のポートに接続されている。この各ポートは固有のＩＤを有している。パケットは、宛先コンポーネントにパケットをルートするために、宛先のル

ータと中間ルータで使用するためのポートＩＤを運ぶ。ルーティング技術の例には、確定的なルーティングを含んでおり、これは、すべてのパケットのためにＡからＢへの同じ経路を選択することを含む。ルーティングのこの形式は、ネットワークの状態とは独立しており、様々なパスの間で負荷分散をしない。これは、基盤となるネットワーク内に存在し得る。しかしながら、そのような決定論的ルーティングはハードウェアで実現することができ、パケットの順序を維持し、ネットワークレベルの停止を無くすことができる。最短パスルーティングは、レイテンシを最小化することができ、そのようなルーティングは送信元から宛先までのホップ数を減少させる。このため、最短経路はまた、２つのコンポーネント間の通信のための最低の電力経路とすることができる。次元順ルーティングは、２－Ｄ、２．５－Ｄ、３－Ｄメッシュネットワークにおけるルーティングの決定論的最短経路の形態である。このルーティング方式では、メッセージが最終目的地に到達するまで、メッセージは特定の順序で各座標に沿ってルーティングされている。例えば、３－Ｄメッシュネットワークにおいて、ルートは、最初にそれが宛先ルータのＸ座標に等しいＸ座標のルータに到達するまで、Ｘ次元に沿ってルーティングされ得る。次に、メッセージは、ターンを行いＹ次元に沿ってルーティングされ、最終的に別のターンを行う。そして、メッセージは、最終的な宛先ルータに到達するまでＺ次元に沿って移動する。ディメンション順ルーティングは、最小ターンと最短パスのルーティングとすることができる。図２（ａ）は、２次元メッシュでＸＹルーティングの例を図示しています。より具体的には、図２は、ノード３４からノード００へのＸＹルーティングを示している。図２（ａ）に示す例のように、各構成要素は、１つのルータの１つのポートのみに接続されている。パケットが、宛先ノードのｘ座標と同じノードのｘ座標であるノード「０４」に到達するまで、パケットは最初にｘ軸の上にルーティングされる。パケットが宛先ノードに到達するまで、パケットは次にｙ軸上にルーティングされる。

【０００７】

１つまたは複数のルータまたは１つ以上のリンクが存在しない、不均一なメッシュトポロジでは、特定の送信元と宛先ノードの間のディメンション順ルーティングが可能でない場合があるため、代替パスを取らなければならない場合がある。その代替パスは、最短または最小ターンではない場合がある。

【０００８】

ソースルーティングとテーブルを使用したルーティングがＮｏＣに使用される他のルーティングオプションである。適応型ルーティングは、ネットワークの状態に基づいて、ネットワーク上の２点間で取られたパスを動的に変更することができる。この形式によるルーティングは、分析して実施するためには複雑である。ＮｏＣ相互接続は、複数の物理ネットワークを含んでいてもよい。各物理ネットワーク上で、複数の仮想ネットワークが存在してもよい。この複数の仮想ネットワークでは、異なるメッセージタイプが異なる仮想ネットワークを介して送信される。この場合、各物理リンクまたはチャネルで、複数の仮想チャネルがある。各仮想チャネルは、両方のエンドポイントに専用のバッファを有することができる。任意の与えられたクロックサイクルにおいて、一の仮想チャネルのみが物理チャネル上でデータを送信することができる。

【０００９】

ＮｏＣ相互接続は、ワームホールルーティングを使用し得る。ワームホールルーティングでは、大きなメッセージまたはパケットは、フリット（フロー制御ディジットともいう）として知られた小片に分割される。最初のフリットは、ヘッダフリットである。これは、このパケットの経路についての情報と、ペイロードデータと共にキーメッセージ・レベルの情報を保持し、以降のすべてのメッセージに関連付けられたフリットのルーティングの動作を設定する。必要に応じて、一つ以上のボディフリットは、データの残りのペイロードを含み、ヘッドフリットに付き従う。最後のフリットは、テイルフリットである。これは、最後のペイロードを含むことに加え、また、メッセージの接続を閉じるためにいくつかの記録を行う。ワームホールフロー制御において、仮想チャネルはよく用いられる。

【００１０】

物理チャネルは、仮想チャネル（ＶＣ）と呼ばれる、多数の独立した論理チャネルにスライスされた時間である。ＶＣｓは、パケットをルーティングするために複数の独立したパスを提供する、しかし、彼らは物理チャネル上で時間多重される。仮想チャネルは、チャネル上のパケットのフリットの取り扱いを調整するために必要な状態を保持する。最低でもこの状態は、ルートの次のホップと、仮想チャネルの状態（アイドル、リソース待ち、またはアクティブ）と、の現在のノードの出力チャネルを識別する。仮想チャネルは、現在のノードと次のノードで使用可能なフリットバッファ数とでバッファリングされているパケットのフリットへのポインタを含むことができる。

【００１１】

用語「ワームホール」は、チャネルを介してメッセージが送信されるように動作する。次のルータの出力ポートは、完全なメッセージが到着する前に、その受信したデータがヘッドフリットの中で変換されることができるように短くすることができる。これにより、ルータがすぐにヘッドフリットの到着時にルートを設定し、その後、会話の残りの部分から離れるようにすることができる。メッセージはフリットによってフリットに送信されるので、異なるルータで、その経路に沿って、メッセージには、いくつかのフリットバッファを占めることができ、ワームのようなイメージを作成する。

【００１２】

様々なエンドポイント間のトラフィックと、各種メッセージのために使用されるルートと物理ネットワークとによって、ＮｏＣ相互接続の異なる物理チャネルは、負荷と輻輳の異なるレベルが発生することがある。ＮｏＣ相互接続の様々な物理チャネルの容量は、チャネル（物理的なワイヤの数）の幅と、それが動作しているクロック周波数によって決定される。ＮｏＣの様々なチャネルは、異なるクロック周波数で動作してもよいし、様々なチャネルは、チャネルにおける帯域幅の要件に基づく異なる幅を有してもよい。チャネルにおける帯域幅の要件は、チャネルとそれらの帯域幅の値の上を横断するフローによって決定される。様々なＮｏＣのチャネル上を横断するフローは、様々なフローにより取られる経路によって影響を受けています。メッシュまたはトーラスのＮｏＣにおいて、等しい長さの複数のルートの経路、または任意の対の送信元ノードと宛先ノードの間のホップ数が存在してもよい。例えば、図２（ｂ）において、ノード３４と００の間の標準ＸＹルートに加えて、ＹＸルート２０３または、送信元から宛先へ複数回ターンするマルチターンルート２０２といった利用可能な追加ルートがある。

【００１３】

様々なトラフィック遅延のために、静的に割り当てられたルートでのＮｏＣでは、インテリジェントに様々なフローのためのルートを選択することにより、種々のチャネルの負荷を制御することができる。多数のトラフィックフローと実質的な経路の多様性とが存在する場合、すべてのＮｏＣチャンネルの負荷がほぼ均一にバランスされるようにルートはそのように選択することができ、これにより、ボトルネックの一点を回避することができる。一度ルーティングされると、ＮｏＣチャネル幅は、チャネル上のフローの帯域幅の要求に基づいて決定され得る。残念ながら、タイミングや配線の煩雑さなどの物理的なハードウェア設計の制限のため、チャネル幅は任意に大きくすることはできない。最大のチャネル幅に制限がある場合があり、これにより、任意の単一のＮｏＣチャネルの最大帯域幅に制限を置いている。

【００１４】

さらに、メッセージが短い場合、広い物理チャネルは、より高い帯域幅を達成するのに役立たないかもしれない。例えば、パケットが６４ビット幅を有する単一のフリットパケットである場合、その後、チャネルが如何に広くても、チャネル上のすべてのパケットが類似している場合、チャネルは、サイクル毎に６４ビットのデータを搬送することができるのみに過ぎない。従って、チャネル幅ものＮｏＣのメッセージサイズによって制限される。最大のＮｏＣチャネル幅のこれらの制限のために、ルートのバランスを取るのにもかかわらず、チャネルは、十分な帯域幅を持っていない可能性がある。

【００１５】

上記帯域幅の問題に対処するために、複数の並列物理ＮｏＣを使用することができる。各ＮｏＣは、レイヤと呼ばれることがあり、このように多層のＮｏＣアーキテクチャを作成する。ホストは、ＮｏＣ層の上にメッセージを注入する、メッセージは、ＮｏＣ層上の宛先にルーティングされる。ここで、メッセージは、ＮｏＣ層からホストへ配信される。このように、各層は、互いに多かれ少なかれ独立して動作し、層の間の相互作用は、注入、吐出時間の間に発生することがある。図３（ａ）は、２つの層のＮｏＣを示している。ここでは、左右の図に複製されたＮｏＣに接続されたホストによって２つのＮｏＣ層が左右に隣接して示されている。ホストは、この例では２つのルータに接続されている。Ｒ１として示した１層目のルータと、Ｒ２として示される２層目のルータとである。この例では、多層ＮＯＣは、３ＤのＮｏＣとは異なる。すなわち、複数の層は、単一のシリコンダイ上にあり、同じシリコンダイ上のホスト間の通信の高帯域幅要求を満たすために使用される。メッセージは、一つの層から別の層に行かない。明瞭化するために、本出願は、このような水平方向の左右の図を利用する。それは、互いの上に垂直にＮｏＣを描くことによって示されている３ＤのＮｏＣと区別するための多層のＮｏＣのためである。

【００１６】

図３（ｂ）において、それぞれＲ１とＲ２との各層からルータに接続されたホストが図示されている。各ルータは他のルータに接続されている、その層において、指向ポート３０１を使用し、注入及び排出口３０２を使用してホストに接続されている。ブリッジ・ロジック３０３は、ホストと２層のＮｏＣ層の間にいることができ、送信メッセージのためのＮｏＣ層を決定する、そして、ＮｏＣ層にホストからメッセージを送信し、また、２のＮｏＣ層からの着信メッセージ間の調停や多重化を実行し、それらをホストに配信する。

【００１７】

多層のＮｏＣでは、必要とされる層数は、システム内に流れるすべてのトラフィックフローの総帯域幅の要件、様々なフローによって使用される経路、メッセージサイズ分布、最大チャネル幅、等のような多くの要因に依存し得る。ＮｏＣ相互接続でのＮｏＣ層数は、設計で一旦決定されると、異なるメッセージとトラフィックフローは、異なるＮｏＣ層上にルーティングされ得る。さらに、異なる層がルータ数、チャネルおよび接続において異なるトポロジを持っているようなＮｏＣ相互接続を設計することができる。異なる層内のチャネルは、チャネルとその帯域幅要件を超えて横断するフローに基づいて、異なる幅を有することができる。

【００１８】

ＮｏＣの相互接続では、トラフィックプロファイルが均一でない場合、そして、一定量の不均一（例えば、特定のホストが他よりも頻繁にお互いに通話している）があると、相互接続性能は、ＮｏＣトポロジに依存し得る、そして、様々なホストは互いに対して、また、それらに接続されているルータに対してトポロジ内に配置される。例えば、二つのホストが頻繁にお互いに通話し、他の相互接続よりも高い帯域幅を必要としている場合、それらは互いに隣接して配置する必要がある。これは、全体の平均レイテンシを減少させるこの通信のためのレイテンシを短縮する、同様にルータノードの数を減少させ、この通信のより高い帯域幅を定められる必要があるものにリンクする。

【００１９】

お互いに近い、動いている２つのホストは遠く離れた特定の他のホストを作り得る、それは、すべてのホストは、互いに重ならない、２Ｄ平面のＮｏＣのトポロジに適合しなければならないからである。このように、さまざまなトレードオフがなされなければならず、特定のグローバルコストと性能測定基準が最適化されるように、ホストは、対の帯域幅とすべてのホスト間のレイテンシの要件を調べた後に配置される必要がある。コストと性能測定基準は、例えば、ルータホップ数において、全ての通信ホスト間の平均構造のレイテンシ、あるいはすべてのホストペア間の帯域幅の合計、そして、ホップ数におけるそれらの間の距離、またはこれら２つの組合せ、であり得る。この最適化問題は、ＮＰ困難であることが知られており、発見的解析に基づいた方法が使用されることが多い。システム内のホストは、互いに対して形状とサイズが変化してもよく、それは、２次元平面のＮｏ

Ｃトポロジでそれらを配置することで、さらなる複雑さを加え、少しの空白を残しながらそれらを最適にパッキングし、そして、重複ホストを回避することができる。
【発明の概要】
【課題を解決するための手段】
【００２０】
本発明の態様は、方法を含み、それは、位置を自動的に決定することを含む。これは、メッシュまたはトーラスネットワークオンチップ（ＮｏＣ）内のさまざまなホストやＩＰコアの相互接続とホストのポートを各種ＮｏＣルータに接続する、ためのものであり、特定の効率関数の形でシステム全体のパフォーマンスが最適化されるように、ルータの数のホップでのＮｏＣ内のすべての通信ホスト間でレイテンシ、帯域幅と距離に関して効率関数を決定し、ホストの位置の結果が互いに空間的にオーバーラップする二つのホストを生じないことを保証し、かつ、異なる形状およびサイズのホスト間の空白を最小化し、すべてのホストのポートが利用可能なＮｏＣルータに接続することができることを保証し、ホスト位置の結果はＮｏＣ内のすべてのホストで必要な接続性に影響を与えないことを保証し、そして、ホストの位置に基づいてＮｏＣルータとパスを自動的に設定する。
【００２１】
本発明の態様は、システムを含み、それは、位置を自動的に決定することを含む。これは、メッシュまたはトーラスネットワークオンチップ（ＮｏＣ）内のさまざまなホストやＩＰコアの相互接続とホストのポートを各種ＮｏＣルータに接続する、ためのものであり、特定の効率関数の形でシステム全体のパフォーマンスが最適化されるように、ルータの数のホップでのＮｏＣ内のすべての通信ホスト間でレイテンシ、帯域幅と距離に関して効率関数を決定し、ホストの位置の結果が互いに空間的にオーバーラップする二つのホストを生じないことを保証し、かつ、異なる形状およびサイズのホスト間の空白を最小化し、すべてのホストのポートが利用可能なＮｏＣルータに接続することができることを保証し、ホスト位置の結果はＮｏＣ内のすべてのホストで必要な接続性に影響を与えないことを保証し、そして、ホストの位置に基づいてＮｏＣルータとパスを自動的に設定する。
【図面の簡単な説明】
【００２２】
【図１ａ】双方向リング、２Ｄメッシュ、２Ｄトーラス、および３ＤメッシュのＮｏＣのトポロジの例を示した図である。
【図１ｂ】双方向リング、２Ｄメッシュ、２Ｄトーラス、および３ＤメッシュのＮｏＣのトポロジの例を示した図である。
【図１ｃ】双方向リング、２Ｄメッシュ、２Ｄトーラス、および３ＤメッシュのＮｏＣのトポロジの例を示した図である。
【図１ｄ】双方向リング、２Ｄメッシュ、２Ｄトーラス、および３ＤメッシュのＮｏＣのトポロジの例を示した図である。
【図２ａ】従来の２次元メッシュでのＸＹルーティングの例を示した図である。
【図２ｂ】送信元と宛先ノードの間の３つの異なる経路を示した図である。
【図３ａ】従来の２層のＮｏＣ相互接続の例を示した図である。
【図３ｂ】ホストと複数のＮｏＣ層の間の従来のブリッジ・ロジックを示した図である。
【図４ａ】システムコンポーネント数およびそれらの間の接続を示した図である。
【図４ｂ】３ｘ６メッシュのＮｏＣトポロジ内のさまざまなホストのサンプル位置を示した図である。
【図４ｃ】３ｘ６メッシュＮｏＣトポロジでのさまざまなホストのより良い位置を示した図である。
【図５】ＮｏＣのトポロジで特定の性能測定基準を改善するためのさまざまなホストを、繰り返し再配置する基準アルゴリズムを示した図である。
【図６】ホストの位置を改善するための、より詳細なアルゴリズムを示した図である。
【図７】ホスト位置計算のためのアルゴリズムのようなシミュレーテッドアニーリング（焼き鈍し法）を実施するために、上記のアルゴリズムを拡張することができる方法を示し

た図である。
【図８】本明細書に記載の実装例を実現することができる、コンピュータ／サーバのブロック図である。
【発明を実施するための形態】
【００２３】
以下の詳細な説明は、本出願の図面及び例の実装の詳細を提供する。参照符号と図の間の冗長要素の説明は、明確にするために省略されている。明細書全体を通して使用される用語は、例として提供され、限定することを意図するものではない。例えば、「自動」という用語の使用は、完全に自動または半自動の実装を含むことができ、本出願の当技術分野の練習の実装で当業者の所望の実装に応じ、実装の特定の側面を制御するユーザまたは管理者に関連する。
【００２４】
複数のルータとルータ間の二点間リンクを使用して、お互いのチップ上でシステムの様々な構成要素を接続し、分配されるＮｏＣ相互接続において、ＮｏＣトポロジで様々なホストのための適切な位置と、その位置でのローカルルータにそれらを接続することを決定する必要がある。例えば、２つのホストが頻繁に互いに通信し、他の配線よりも高い帯域幅を必要とする場合、それはお互いにそれらをより近く配置するほうがよい。それは、これらのホスト間のやり取りがより少ないルータホップとリンクの上で行うことができ、そして、全体的なレイテンシとＮｏＣコストを低減することができるようにするためである。
【００２５】
特定の形や大きさを持つ２つのホストが互いに２ＤのＳｏＣ平面上で空間的に重複しないと仮定すると、トレードオフが必要になる場合がある。それらの間の相互通信を改善するために、近い特定のホストを移動することは、特定の他のホストを強制的にさらに離れさせ得る、それによって、それらの他のホストとの間の相互通信を不利にする。システムのパフォーマンスを向上させるトレードオフを行うために、特定のシステム性能測定基準は、ホストがＮｏＣのトポロジ内に配置されるよう最適化するために、例えば全体の平均通信レイテンシを目的関数として使用することができる。システムの性能測定基準を最大化する、実質的に最適なホスト位置を決定することは、すべてのホストの間の接続と相互通信特性の分析と、慎重に２ＤのＮｏＣトポロジにそれらを配置することとを伴う。
【００２６】
３ｘ６メッシュ組織に配置されるべき１６個のＣＰＵと２個のメモリがある、という例を考える。図４（ａ）に示されるように、第１のメモリＭＥＭ１と通信する第１の８個のＣＰＵセットと、第２のメモリＭＥＭ２と通信する第２の８個のＣＰＵセットとを仮定する。図４（ｂ）に示されように、ＣＰＵとメモリは、順番に３ｘ６メッシュ内に配置されてもよい。各ホストは、メッシュ内のセルを占めており、直接セルのルータに接続されている。ここでは、さまざまなホスト間のトラフィックを考慮しない。
【００２７】
相互通信ホストは、互いに遠くに配置されており、これにより、ホップ数において平均値を高くし、かつ、構造的なレイテンシのピークに導く。例えば、図に示すように、ホストＣＰＵ１とＭＥＭ１との間のメッセージは、７ルータノードを移動する必要があり、ホストＣＰＵ１３とＭＥＭ２との間のメッセージは、６ホップを移動する必要がある。このような長いパスは、レイテンシを増加させるだけでなくだけでなく、相互接続帯域幅に悪影響を与える、即ち、メッセージが長い期間ＮｏＣに留まり、多数のリンクの帯域幅を消費する。
【００２８】
図４（ｃ）に示すように、一例として、上記のホストのセットは組織内に置いてもよい。これは、有意に平均と構造的なレイテンシのピーク値を減少させる。図示されるように、この組織の相互通信ホスト間の最大の構造的なレイテンシは、３ルータのホップであり、相互通信ホストの大部分は、２ルータのホップが離れている。以前の位置から新しい位

置に、特定のホストを繰り返し再配置し、その過程で、新しい位置に既に存在するホストはそれらを交換することにより、図４（ｃ）中のホスト位置は、図４（ｂ）から実現することができる。

【００２９】

最適なホスト位置を思い付くために、この例は比較的直感的であるが、トラフィックプロファイルが複雑な接続性と、高い非対称帯域幅と、さまざまなホスト間のレイテンシの仕様とで構成されている場合は、ＮｏＣトポロジ内のホストに最適な位置を決定することは、さらに困難となり得る。実際には、既知のＮＰ困難問題に帰着することができる。このように、発見的解決法は、このような設定によって最適なホストの位置を決定するために使用されるべきである。本明細書に記載された実装例は、特定のシステム性能測定基準を最適化するＮｏＣトポロジで、さまざまなホストを配置するために、多数の発見的アルゴリズムを説明する。

【００３０】

提案された発見的問題解決法では、すべてのホスト間の接続、帯域幅とレイテンシの要件、そして、その形状およびサイズは、ホストの位置を決定するために全体的なレベルで分析される。本明細書に記載された実装例は、システムのＮｏＣ相互接続性能を向上させるために、それらの形状およびサイズと間の通信特性に基づき、ＮｏＣのトポロジでのさまざまなホストの位置の自動計算に利用される、２－Ｄ、２．５Ｄおよび３ＤのＮｏＣ相互接続のための解決策に向けられており、ＮｏＣコストを削減する。実装例は、以下のものに関連する。

【００３１】

１）ＮｏＣトポロジ内のホスト位置に基づいてコスト関数、様々なやり取りが取られたルート、ホスト間の通信帯域幅とレイテンシの仕様、そして、所望の性能測定基準を構築する；

２）初期位置にホストを配置し、ローカルのＮｏＣルータにそれらを接続し、すべてのやり取りのルートを計算し、初期費用関数を計算する。

３）現在位置Ｐ１にホストｈ１を選択し、ＮｏＣのホストのための新たに取りうる位置ｐ２を求める。

４）位置ｐ２にホストｈ１の再配置方法を決定する、すなわち、ｐ２でのホストの現在のセットでそれを交換することは、コスト関数に影響を及ぼし得る。

５）上記の移転を受け入れるか、拒否するかどうかを決定する。

６）特定の基準が満たされるまで、最終的にステップ３，４、および５を繰り返す。

【００３２】

実装例は、図５に記載されている。５００では、ホストはＮｏＣトポロジで初期位置に配置されており、ＮｏＣのローカルルータに接続されている。現在の位置、接続およびホスト間通信の特性に基づいて、費用関数が計算される。次に、５０１で示されるように、新しい位置への移転のため、現在位置ｐ１でホストｈ１が選択される。移転のためのホストを選択するために、多くの基準が使用され得る。そのうちのいくつかは、後述する追加の実装例に記載されている。

【００３３】

５０２において、新しい位置ｐ２は、ホストｈ１のために選択される。ホストｈ２がｐ２に既に存在している場合、ｈ１が再配置されると、ｈ２はｈ１と交換される。５０３において、ｈ１がｐ２に再配置される場合は、新しいコスト関数が計算される（そして、ｈ２　をｐ１に、もしｈ２が存在する場合）。５０４において、新しいコスト値と古いコスト値及び受付関数に基づいて、ｈ１とｈ２が再配置されているかどうか判断される（相互に入れ替える）。最後に、５０５において、特定の基準が満たされているか否か、または特定の最大反復回数に達しているかどうかということに基づいて、ステップ５０１で継続するか否かを判定する。

【００３４】

所望の実装に応じて、最大反復回数は、多くの方法で決定することができる。例えば、

反復する多数のプロセスを実行し、最大反復回数が決定され得る。また、これまでに見つかった最小のシステム・コストが安定したとき、そして、低コストの新たなシステムの位置が試行多数の後に発見されなかったときに調べる。この回数は、将来の最適化のための最大反復回数として使用することができる。６６×ｎ３の最大反復回数値は、ｎ×ｎのメッシュネットワークのための実験において、一般的に良好な結果を与える。

【００３５】

しかし、より高い値は、最適化の高いエフォートモードで使用することができる。お互いに複数のホストのグループを交換するために、別の実装例を試みることができる。ホストは、形や大きさが異なる場合には、新しい位置に大きなホスト用のスペースを確保するために、大きいホストは複数の小さなホストと交換され得る。そして、その逆も同様である。

【００３６】

コスト関数は、多くの方法で考案することができる。単純な費用関数は、多数のルータのホップ内におけるすべてのホストのペア間の距離の積の和、およびそれらの間の通信帯域幅要件であってもよい。これは、ＮｏＣのリンクが経験するトラフィック負荷の総量を反映する。ｈｉがｉ番目のホストを表す場合、帯域幅（ｈｉ，　ｈｊ）は、ホストｈｉからホストｈｊまでの帯域幅を表し、距離（ｈｉ，　ｈｊ）は、多数のルータホップにおける、ホストｈｉホストｈｊまでの距離を表し、コスト関数は以下のようになる。

cost = Σ bandwidth(hi, hj) × distance(hi, hj) for all i, j, and i ≠ j

このコスト関数は、最小化した場合、ＮｏＣのチャンネルに必要な帯域幅を最小限に抑えることができる。それによって、ＮｏＣのコストを最小限に抑えることができる。

【００３７】

これにより、ＮｏＣ距離はまた、構造的なレイテンシを表すことができ、コスト関数は、システム内の全体的な構造的レイテンシの平均を最小化することができる。距離及びレイテンシがＮｏＣトポロジにおいて強く相関しない場合、代替コスト関数の実装も関数内の別の要因として、ｈｉとｈｊからレイテンシを使用してもよい。特定のホストとの間の相互通信が他の相互接続よりも重要である場合には、互いに関して様々なやり取りの重要性を反映するために、コスト関数の実装では、関数内の重みを使用することができる。

cost = Σ bandwidth(hi, hj) × distance(hi, hj) × weight(hi, hj) for all i, j, and i ≠ j

コスト関数のいくつかの他の形態もまた、最適化の目標に基づいて使用することができる。

【００３８】

ＮｏＣトポロジのホストのための初期位置は、いくつかの方法で選択され得る。実装例は、どの２つのホストも空間的に重ならないように、ランダムにホストを置くことができる。そして、ローカルルータのホストポートに、ホストのポートを接続する。特定のホストは、単一のメッシュセルよりも大きくてもよく、ＮｏＣの特定のルータおよびチャネルを遮断することで、特定のルートを防止することができる。ホストは、最初に配置する必要がある。それは、すべての相互通信のホストが少なくともそれらの間に利用可能な１つのＮｏＣルートを持ち、それらを互いに通信可能にするためである。

【００３９】

ホストの再配置時に、再配置はＮｏＣ中のすべてのやり取りのルーティング可能性を維持するものに限定されるべきである。他の実装例では、いかなる輻輳が発生することを避けるため、初期位置にホストを置くことができる。すなわち、これらの位置では、もはやどの２つのホストも、互いに交換することはできない。最小限の制限で自由に発生するホストの交換を可能とする初期位置は、好ましい場合がある。

【００４０】

他の実装例は、移転のための特定のホストを選択すること、または他のホストより頻繁に交換すること、をさらに含んでもよい。さらに、アルゴリズムが実行する回数は、内側と外側の２つのループを形成する２つの反復カウンタによって制御することができます。

内側のループでは、ホストは、再配置のために選択することができ、再配置は、各パス内の特定の基準に基づいて受け入れてもよい。外側のループでは、各パスでこれらの基準関数を調整してもよい。

【００４１】

このような設計は、図６で説明されている。６００において、初期位置にホストを入れた後、外側ループの反復カウンタｉが０に設定されている。６０１で、内側ループの反復カウンタｊは０に設定されており、さまざまなホストに割り当てられた重みに基づいて、ホストｈ１は、再配置の検討のため選択される。再配置検討のためのホストへの重み付けは、さまざまなホストのレイテンシと帯域幅の仕様に依存してもよい。一例として、少数のホストと通信するものより頻繁に再配置のために、多数のホストと通信するようなホストを考慮することができる。例えば、重みはホストに割り当てられ、ｈｉは以下のようになり得る。

weight(hi) = bandwidth(hi) / Σ bandwidth(hi)

一つは、内側と外側のループの反復カウンタで重みへ同様に分解することができ、アルゴリズムが進行するにつれ、より良いホストの選択を制御する。

【００４２】

次に、ホストｈ１の新しい位置が決定される。６０２において、ｈ１と通信するすべてのホストが、検討されている。それらの一つであるｈ２は、一定の確率関数ｆ１に基づいて選択される（例えば、時間のｘ％でホストｈ２を選択、ここで、ｘは、確率関数で示される確率、等）。確率関数を使用すると、ホストｈ１のための新しい位置を決定する際に、いくつかのランダム性を取り込む、このとき、極小値を回避することが重要である。次に、ステップ６０３において、ホストｈ２のすべての隣接位置が検査され、それらの一つであるｐ３は、確率関数ｆ２に基づいて選択される。

【００４３】

これは、ホストｈ１のための新しい位置である。次に、ステップ６０４において、ｈ１がｐ３を位置決めするために再配置される場合、および位置ｐ３にホストｈ３の存在があった場合に新しいコスト関数が計算される、そして、ホストｈ３はｈ１の現在の場所に移転される。新しいコスト、以前のコスト、および確率受付関数ｆ３に基づいて、再配置は、再配置の決定に基づいて承認または拒否される（例えば、時間のｘ％を受け入れる。ここで、ｘは確率関数によって示された確率。等）。次に、最大反復回数に対して内側と外側のループの反復カウンタが更新され、比較される、そして、図に示すように、プロセスが繰り返される。外側のループの各パスの間に、ステップ６０５において、関数ｆ１、ｆ２と受け入れの関数ｆ３は調整され得る。

【００４４】

関数ｆ１は、ｈ１と、より頻繁に通信するこれらのノードを優先するように構築され得る。ホストｈ１がｈ１＿１からｈ１＿ｎという名のｎ個のホストに通話する場合、これらのホスト間の通信帯域幅の要件は、Ｂ（ｈ１、ｈ１＿ｉ）、ｉ　＝１からｎであり、そして、ホストｈ２がｈ１＿１からｈ１＿ｎの中から選択される確率は、以下の式で与えられる。

P(h1_i) = B(h1, h1_i) / Σ B(h1, h1_i)

同様に、レイテンシの要件があり得、特定のホストは他のものよりも低いレイテンシでｈ１と通信可能でなければならない。

【００４５】

Ｌ（ｈ１、ｈ１＿ｉ）、ｉ＝１からｎまで、という条件で与えられるようなレイテンシ制約がある場合は、以下に示すように、帯域幅およびレイテンシの両方を調整して、確率関数を適合させてもよい。

P(h1_i) = (B(h1, h1_i) / Σ B(h1, h1_i)) × (Σ L(h1, h1_i) / L(h1, h1_i))

一つの加重確率関数を使用することができ、これで帯域幅とレイテンシは、与えられたシステムにおいて、その重要性を反映した異なる重み値が与えられる。

【００４６】

関数ｆ２は、多くの方法で考案され得る。シンプルなデザインは、均一な確率により、すべての直接に隣接するものの中で、ｈ２と直接に隣接するものを使用することができる。優先権は、あまり頻繁にこのｈ２と通話しない隣接するものに与えられてもよい。これにより、頻繁にｈ２と通話するホストがｈ２から離れて移動することを回避する。これは、ｈ２およびその隣接するホストの帯域幅とレイテンシ情報を使用して確率関数に反映させてもよい。

【００４７】

関数ｆ３は、多くの方法で考案され得る。例えば、再配置がコスト関数を減少させる場合には、再配置は常に受け付けてもよい。一方、移転がコストを増加させる場合には、一定の確率関数によって再配置が受け入れられる。確率関数は、最適化のタイプに基づいて考案され得る。例えば、焼き鈍し法に基づく手法で、受理確率は「温度」と呼ばれる時間的に変化するパラメータに依存し得る、これはコスト関数を増加させ、再配置を受諾するか拒否するかを制御する。アルゴリズムは、高い温度で内側のループを反復することを開始する、そして、内側ループの各パスで次第に温度を低下させる。

【００４８】

より高い温度レベルで費用関数を増加させる再配置は、より低い温度レベルにおける場合に比して、より高い確率で認められている。このようにして、より高い温度で極小値から遠ざかることについて、より高い可能性を許容する。システムが冷却されるように、（すなわち、温度が低下する）、コスト関数を増やす動きが低い確率で受け入れられている。各外側反復ループの開始時に、温度は、より高いレベルで復元することができる。このように、内側のループで冷却プロセスの複数のパスを作成する。

【００４９】

関数ｆ３は、次のように定義することができる。

P(relocation) = 1/(1 + exp((cost(old) - cost(new)) / cost(initial)×temp))

ここで、Ｐ（ｒｅｌｏｃａｔｉｏｎ）は、再配置を受け付けたか否かの確率を示す。ｃｏｓｔ（ｏｌｄ）は、移転前のコスト関数であり、ｃｏｓｔ（ｎｅｗ）は、移転後のコスト関数である。ｃｏｓｔ（ｉｎｉｔｉａｌ）は、システムの初期費用関数であり、ｔｅｍｐは、現在の温度レベルである。初期温度レベルを１またはいくつかの他の値に選択することができ、内側のループのパスのすべての固定数の後に等比数列で減少する。温度は、同様に他の方法で低減することができる。内側のループは、温度がいくつかの予め定められた最低許容値Ｔ（ｍｉｎ）に達するまで継続することができる。

【００５０】

この手順は、図７に記載されている。図６に示される実施例と比較して、この実装では、ステップ７００で、各内側ループの開始時に、温度を初期化する。ステップ７０１で、幾何学的なシーケンスに従って、それを減少させる。温度が最小に設定された値に達すると、内部ループが停止する。関数ｆ１とｆ２も、現在の温度レベル、ホスト間通信の帯域幅とレイテンシの仕様およびさまざまなホストの現在位置に基づいて、定義することができる。典型的な実装例は、１である初期温度Ｔ（ｉｎｉｔｉａｌ）と０．００１である最終温度Ｔ（ｍｉｎ）とを使用することができる。

【００５１】

内部ループのｉ番目のパスの間、現在の温度Ｔは、以下の式に従って更新されてもよい。

T = T(initial) x (0.9)(i/C)

ここで、Ｃは、ネットワークおよびホストの数の大きさに応じたシステム定数である。

C = n3,

ここで、ｎは、ホストの総数であり、特定の例示的な実装形態において良好な結果を与える。温度Ｔが最終値Ｔ（ｍｉｎ）に達すると、その後内部ループを終了する。

【００５２】

関数ｆ１、ｆ２、およびｆ３はまた、標準的な最適化手法の動作をエミュレートするための、遺伝的アルゴリズムや機械学習などの様々な実装例による、追加のいくつかの方法

で考案されてもよい。
【００５３】
図８は、その上に実装例を実装することができる例示的なコンピュータシステム８００を示している。当業者に知られているように、コンピュータシステム８００は、１つ以上のユニットを実行するためにＩ／Ｏユニット８３５、記憶装置８６０、および動作可能なプロセッサ８１０を有することができるサーバ８０５を含む。
【００５４】
本明細書で使用する用語「コンピュータ可読媒体」は、実行のためにプロセッサ８１０に命令を提供することに関与する任意の媒体を指し、これは、コンピュータ可読記憶媒体の形となることがあり、例えば、光ディスク、磁気ディスク、リードオンリーメモリ、ランダム・アクセス・メモリ、ソリッドステート装置及びドライブに限らず、または電子情報を格納するのに適した有形の媒体の任意の他のタイプ、またはコンピュータ可読信号媒体、であり、これらはキャリアウェーブのような一時的な媒体を含むことができる。Ｉ／Ｏユニットは、ユーザインターフェース８４０とオペレータインタフェース８４５からの入力を処理し、これは、キーボード、マウス、タッチデバイス、または言葉によるコマンドなどの入力装置を利用することができる。
【００５５】
サーバ８０５はまた、外部記憶装置８５０に接続することができる。これはリムーバブルストレージを含めることができる。例えば、ポータブルハードドライブ、光メディア（ＣＤまたはＤＶＤ）、ディスク媒体またはコンピュータが実行可能なコードを読み取ることができる任意の他の媒体である。サーバはまた、出力装置８５５、例えば、ユーザにデータおよび他の情報を出力し、同様に、ユーザからの追加情報を要求するディスプレイを接続することができる。
【００５６】
サーバ８０５から、ユーザインターフェース８４０、オペレータインタフェース８４５、外部記憶装置８５０、出力装置８５５への接続は、無線プロトコルを介してもよい。例えば、８０２．１１規格、Ｂｌｕｅｔｏｏｔｈ（登録商標）またはセルラープロトコル、または、ケーブルや光ファイバなどの物理的な伝送媒体を介したものである。出力装置８５５は、したがって、ユーザと対話するための入力装置として機能することができる。
【００５７】
プロセッサ８１０は、一つ以上のモジュールを実行してもよい。任意のホストの重複を回避し、また、必要な接続が成立していることを保証する２次元平面内において、いくつかの初期の位置でホストを配置するために初期配置モジュール８１１を構成することができる。コスト関数モジュール８１２は、トポロジ内のＮｏＣホスト位置、様々なやり取りが行われるルート、ホスト間の通信帯域幅とレイテンシの仕様、構造的なレイテンシなどの所望の性能測定基準、配線部分や電源などに基づいてコスト関数を構築する。再配置ホスト選択モジュール８１３は、新しい位置に再配置するために検討されているさまざまなホスト、および対応する新たな位置を選択するように構成されてもよい。
【００５８】
再配置受付関数モジュール８１４は、再配置ホスト選択モジュールからの再配置の決定を受け入れるか、新しいホスト位置のコストを決定するコスト関数モジュールからのフィードバックに基づいて拒否するか、を決定する。再配置ホスト選択モジュール８１３と、再配置受付関数モジュール８１４とは、上述したように例示的な実装形態に基づいて実装されてもよい、例えば、図５－７に示されるものである。
【００５９】
また、詳細な説明のいくつかの部分は、コンピュータ内のオペレーションのアルゴリズム及び記号表現に関して提示される。これらのアルゴリズム記述および記号表現は、最も効果的に他の当業者に彼らの技術革新の本質を伝えるために、データ処理分野の当業者によって使用される手段である。アルゴリズムは、望ましい最終状態又は結果に導く一連の定義されたステップである。実装例で実行されるステップは、有形の結果を達成するため

の具体的な量の物理的操作を必要とする。

【００６０】

また、本発明の他の実装は、本明細書に開示の例示的な実装の仕様および実施を考慮すれば当業者にとって明らかであろう。様々な態様および／または記載された例の実装の構成要素は、単独で又は任意の組み合わせによって使用することができる。明細書および実施例は、アプリケーションの真の範囲および精神が以下の特許請求の範囲によって示される例として考慮されることが意図されている。