

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5285083号
(P5285083)

(45) 発行日 平成25年9月11日(2013.9.11)

(24) 登録日 平成25年6月7日(2013.6.7)

(51) Int.Cl.

H04L 12/70 (2013.01)

F 1

H04L 12/56 400Z

請求項の数 17 (全 16 頁)

(21) 出願番号 特願2010-536836 (P2010-536836)
 (86) (22) 出願日 平成20年11月21日 (2008.11.21)
 (65) 公表番号 特表2011-505778 (P2011-505778A)
 (43) 公表日 平成23年2月24日 (2011.2.24)
 (86) 國際出願番号 PCT/KR2008/006861
 (87) 國際公開番号 WO2009/072768
 (87) 國際公開日 平成21年6月11日 (2009.6.11)
 審査請求日 平成23年9月8日 (2011.9.8)
 (31) 優先権主張番号 11/949,167
 (32) 優先日 平成19年12月3日 (2007.12.3)
 (33) 優先権主張国 米国(US)

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

最終頁に続く

(54) 【発明の名称】並行してトポロジを発見する方法および装置

(57) 【特許請求の範囲】

【請求項 1】

ネットワークのトポロジを発見する、コンピュータ実装可能な方法であって、前記方法は、ノード・エントリの第1リストを受信するステップであって、前記ノード・エントリの各々は既知ノードを表す一意のノード識別子を含む、受信するステップと、前記ノード・エントリを複数のノード・グループに分割するために、前記ノード・エントリの前記一意のノード識別子に演算を施すステップであって、前記一意のノード識別子にハッシュ演算が施されてハッシュ値が生成される、演算を施すステップと、

第1ノード発見エージェントを第1ノード・グループに割り当て、第2ノード発見エージェントを第2ノード・グループに割り当てるステップであって、前記第1および第2ノード発見エージェントは、ノードからノード情報を収集するように動作可能である、割り当てるステップと、

第1ノードからノード情報を収集するステップであって、前記ノード情報は、前記第1ノード発見エージェントによって収集され、前記第1ノードと接続されるノードとの間の接続を記述する、収集するステップと、

前記ノード情報を、ノード・エントリの第2リストに付け加えるステップと、ネットワーク・トポロジを生成するべく、ノード・エントリの前記第2リストを処理するステップと、

10

20

を含む、方法。

【請求項 2】

前記第2ノードの前記ノード識別子が前記第1リストに記載されているかどうかを判断するステップと、

記載されていない場合には、前記第2ノードの、ノード・グループへの割り当てを特定するため、前記第2ノードの前記一意のノード識別子に演算を施すステップと、

前記特定されたノード・グループに前記第2ノードを付け加えるステップと、

前記第2ノードからノード情報を収集するステップであって、前記ノード情報は、前記特定されたノード・グループに割り当てられている前記ノード発見エージェントによって収集され、前記第2ノードと第3ノードとの間の接続を記述する、収集するステップと、
10
をさらに含む、請求項1に記載の方法。

【請求項 3】

前記第1ノードと第2ノードとの間の前記接続は物理的である、請求項1に記載の方法。

【請求項 4】

前記第1ノードと第2ノードとの間の前記接続は論理的である、請求項1に記載の方法。

【請求項 5】

前記第1ノードと第2ノードとの間の第1接続は物理的であり、第2接続は論理的である、
請求項1に記載の方法。

【請求項 6】

プロセッサと、
20

前記プロセッサに結合されているデータ・バスと、

コンピュータ・プログラム・コードを組み込んでいるコンピュータ使用可能媒体であって、前記コンピュータ使用可能媒体は、前記データ・バスに結合されており、前記コンピュータ・プログラム・コードは、ネットワークのトポロジを発見し、さらに、前記プロセッサによって実行可能な命令を含み、前記命令は、以下のとく：

ノード・エントリの第1リストを受信することであって、前記ノード・エントリの各々は既知ノードを表す一意のノード識別子を含む、受信すること；

前記ノード・エントリを複数のノード・グループに分割するために、前記ノード・エントリの前記一意のノード識別子に演算を施すことであって、
30
前記一意のノード識別子にハッシュ演算が施されてハッシュ値が生成される、演算を施すこと；

第1ノード発見エージェントを第1ノード・グループに割り当て、第2ノード発見エージェントを第2ノード・グループに割り当てることであって、前記第1および第2ノード発見エージェントは、ノードからノード情報を収集するように動作可能である、割り当てること；

第1ノードからノード情報を収集することであって、前記ノード情報は、前記第1ノード発見エージェントによって収集され、前記第1ノードと接続されるノードとの間の接続を記述する、収集すること；

前記ノード情報をノード・エントリの第2リストに付け加えること；および

ネットワーク・トポロジを生成するべく、ノード・エントリの前記第2リストを処理すること
40

を行うように構成されている、コンピュータ使用可能媒体と、
を含む、システム。

【請求項 7】

前記第2ノードの前記ノード識別子が前記第1リストに記載されているかどうかを判断すること；

記載されていない場合には、前記第2ノードの、ノード・グループへの割り当てを特定するべく、前記第2ノードの前記一意のノード識別子に演算を施すこと；

前記第2ノードを前記特定されたノード・グループに付け加えること；および

前記第2ノードからノード情報を収集することであって、前記ノード情報は、前記特定さ
50

れたノード・グループに割り当てられている前記ノード発見エージェントによって収集され、前記第2ノードと第3ノードとの間の接続を記述する、収集することをさらに含む、請求項6に記載のシステム。

【請求項8】

前記第1ノードと第2ノードとの間の前記接続は物理的である、請求項6に記載のシステム。

【請求項9】

前記第1ノードと第2ノードとの間の前記接続は論理的である、請求項6に記載のシステム。

【請求項10】

前記第1ノードと第2ノードとの間の第1接続は物理的であり、第2接続は論理的である、請求項6に記載のシステム。

【請求項11】

コンピュータ・プログラム・コードを組み込んでいるコンピュータ使用可能媒体であって、前記コンピュータ・プログラム・コードは、コンピュータ実行可能命令を含み、前記命令は、以下のステップ：

ノード・エントリの第1リストを受信するステップであって、前記ノード・エントリの各々は既知ノードを表す一意のノード識別子を含む、受信するステップ；

前記ノード・エントリを複数のノード・グループに分割するために、前記ノード・エントリの前記一意のノード識別子に演算を施すステップであって、

前記一意のノード識別子にハッシュ演算が施されてハッシュ値が生成される、演算を施すステップ；

第1ノード発見エージェントを第1ノード・グループに割り当て、第2ノード発見エージェントを第2ノード・グループに割り当てるステップであって、前記第1および第2ノード発見エージェントは、ノードからノード情報を収集するように動作可能である、割り当てるステップ；

第1ノードからノード情報を収集するステップであって、前記ノード情報は、前記第1ノード発見エージェントによって収集され、前記第1ノードと接続されるノードとの間の接続を記述する、収集するステップ；

前記ノード情報をノード・エントリの第2リストに付け加えるステップ；およびネットワーク・トポロジを生成するべく、ノード・エントリの前記第2リストを処理するステップ

をコンピュータに実行せしめるように構成されている、コンピュータ使用可能媒体。

【請求項12】

前記第2ノードの前記ノード識別子が前記第1リストに記載されているかどうかを判断するステップ；

記載されていない場合には、前記第2ノードの、ノード・グループへの割り当てを特定するべく、前記第2ノードの前記一意のノード識別子に演算を施すステップ；

前記第2ノードを前記特定されたノード・グループに付け加えるステップ；および前記第2ノードからノード情報を収集するステップであって、前記ノード情報は、前記特定されたノード・グループに割り当てられた前記ノード発見エージェントによって収集され、前記第2ノードと第3ノードとの間の接続を記述する、収集するステップ

をさらに含む、請求項11に記載のコンピュータ使用可能媒体。

【請求項13】

前記第1ノードと第2ノードとの間の前記接続は物理的である、請求項11に記載のコンピュータ使用可能媒体。

【請求項14】

前記第1ノードと第2ノードとの間の前記接続は論理的である、請求項11に記載のコンピュータ使用可能媒体。

【請求項15】

10

20

30

40

50

前記第1ノードと第2ノードとの間の第1接続は物理的であり、第2接続は論理的である、請求項1_1に記載のコンピュータ使用可能媒体。

【請求項1_6】

前記コンピュータ実行可能命令は、リモート位置にあるサーバからクライアント・コンピュータへ配置可能である、請求項1_1に記載のコンピュータ使用可能媒体。

【請求項1_7】

前記コンピュータ実行可能命令は、サービス・プロバイダによって顧客へオンデマンド・ベースで提供される、請求項1_1に記載のコンピュータ使用可能媒体。

【発明の詳細な説明】

【技術分野】

10

【0001】

本開示の実施形態は、全般的に、コンピュータおよび類似技術の分野に関し、特に、この分野で利用されるソフトウェアに関する。さらに具体的には、本開示の実施形態は、ネットワークのトポロジを発見することに関する。

【背景技術】

【0002】

インターネットの発達は、近年におけるストリーミング映像、ストリーミング音声などといったアプリケーションの普及ならびにヴォイス・オーバー・インターネット・プロトコル(VOIP: voice over Internet protocol)トラフィックの大規模な集積と相まって、ネットワークの帯域幅および可用性への需要を高めてきた。こうした需要を満たすために、追加的なスイッチ、ブリッジ、およびハブが導入されているのであるが、その数は増える一方である。結果として、管理対象となるネットワーク・インフラ量が増加し続けることになる。ネットワーク要素の効果的な管理には、ネットワークのノード群とノード相互間の接続とを表現したものであるネットワーク・トポロジに関する、正確かつ最新の知識が必要になる。ネットワーク要素のトポロジに関する知識は、一般に、事後対応型および事前対応型資源管理、性能評価および最適化、フロー制御、イベント相関、およびサービス障害の根本的原因解析などといった多くのネットワーク管理タスクに不可欠なものである。ところが、ネットワークのサイズが大きくなるにつれて、ネットワーク・トポロジに関する情報を発見することがいっそう困難になる。

20

【0003】

30

ネットワーク・トポロジを発見する手法として、ネットワーク内の各ノードならびにそのノードと他ノードとの接続に関する情報を収集しながらネットワーク要素のグラフ・トラバーサルを行う、発見エージェントを使用することが挙げられる。発見プロセスは、上記のエージェントが、ネットワーク内の一連の既知ノードを与えられることから始まる。例えばシンプル・ネットワーク管理プロトコル(SNMP:simple network management protocol)クエリなどのプローブを用いて、あるいはルーティング・テーブル情報を読み取ることによって、エージェントは、グラフの既知の部分から流れ出すリンクを発見する。エージェントは次に、そのリンクの反対側にあるノードを見つける。このプロセスは、リンクが一つも残らず探索されるまで続く。

【発明の概要】

40

【発明が解決しようとする課題】

【0004】

上記の発見プロセスには、主要な性能指標が2つある。1つ目は、ネットワークを発見するのにかかる時間であり、2つ目は、ネットワーク・ノードへ送信されるプローブ・メッセージの数である。最も効率のよい発見プロセスであれば、プローブ・メッセージは1ノードにつき1つしか送信されないのであろう。発見エージェントが1つの場合、ネットワークを発見するのにかかる時間とは、各プローブ・メッセージに対する応答時間を合計したものである。何千あるいは何万ものネットワーク・ノード群から成る大規模ネットワークでは、上述のようなグラフ・トラバーサルにかなりの時間が費やされる可能性がある。複数のエージェントが、並列にグラフを発見しトラバースするように並列に動作すること

50

が可能であれば、グラフ・トラバーサルを行うのに必要な時間を減らすことができる。一方で、そのように並列して発見することは、個別のノード1つ1つに向けて複数のプローブ・メッセージを送信するという結果になりかねない。このため、複数のノード発見エージェントが並列に機能することは、適切な調整が行われない限り、ネットワーク・トポロジの発見に費やされる時間を減らすという成果をもたらさない。

【課題を解決するための手段】

【0005】

本発明は、並列に動作する複数のノード発見エージェントを用いたネットワーク・ノード発見マネージャによってネットワークのトポロジを発見する方法、システム、およびコンピュータ使用可能媒体を含むが、これらに限定されない。多様な実施形態において、ネットワーク・ノード発見マネージャは、ターゲット・ネットワーク内の既知ノード各々の一意のノード識別子を含んだNodeListを取得する。次に、ターゲット・ネットワークのトポロジを発見するべく並列に用いられるノード発見エージェントの数（「K」）が決められる。ノード発見エージェントについては、1つから既知ノードの総数に至るまで、任意の数のノード発見エージェントを選択することができる（すなわち、1 K ノードの総数）。既知ノードおよび発見されたノードの双方に関するノード情報を保持するために、ネットワーク・ノードのGraphListがノード発見マネージャによって作成され、最初は、空に指定される。

10

【0006】

既知ノード群を、ノード発見エージェントの数（「K」）と同数のノード・グループに分割するために、NodeList内の一意のノード識別子にハッシュ演算が施される。各ノード・グループに関連している生成ハッシュ値は、その後、NodeList内の対応する一意のノード識別子に付け加えられる。ノード・グループに関連しているハッシュ値は、次いでノード発見エージェント「M」にそれぞれ割り当てられる。このとき、 $1 \leq M \leq K$ である。ノード発見マネージャによって、GraphListのノード情報受け取りスレッドが開始され、続いてノード発見エージェント「M」各々のノード発見スレッドが開始される。次に、未処理のノード識別子エントリがNodeListからポップされ、対応するノード・グループ・ハッシュ値が特定される。そのノード・グループ・ハッシュ値に割り当てられているノード発見エージェント「M」がビジーである場合、利用可能なノード発見エージェント「M」に対応するノード識別子エントリがポップされるまで上記のプロセスが繰り返される。

20

【0007】

ノード発見エージェント「M」は、その後、SNMPクエリなどのプローブ・メッセージを、ターゲットの既知ノードへ送信する。プローブにより、ターゲット・ノードは周辺ノードに接続されていると判断された場合、周辺ノードとの接続に関する情報が収集される。一実施形態では、接続は物理的接続である。別の実施形態では、接続は論理的である。さらに別の実施形態では、接続は物理的もしくは論理的接続の組み合わせであってもよい。収集されたノードおよび接続の情報が処理されて、発見された周辺ノードの一意のノード識別子が特定される。発見された周辺ノードの一意のノード識別子がNodeListに現在記載されていない場合には、それらの一意のノード識別子にハッシュ演算が施される。次に、結果として生じたグループ・ノード・ハッシュ値を用いて、発見された周辺ノードをノード・グループに割り当て、対応するノード発見エージェント「M」による処理を行う。ターゲット・ノードと発見された周辺ノードとの接続がGraphListに現在記載されていない場合には、これらの接続が追加される。このプロセスは、「K」個のノード発見エージェント「M」全てが、既知ノードおよび発見された周辺ノードに関する各自のノード発見動作を完了するまで、継続される。「K」個のノード発見エージェント「M」によって全てのノード情報が収集されてGraphListに記載されたら、ネットワーク・トポロジが生成される。上記のことに加え本発明の追加的な目的、特徴および利点が、以下に詳細に記載された説明の中で明らかとなるだろう。

30

【0008】

40

50

以下の詳細な記載を添付の図面とともに考察すると、本発明の選ばれた実施形態について、および本発明の数々の目的、特徴および得られる利点について、理解することができる。

【図面の簡単な説明】

【0009】

【図1】本発明を実装し得る例示的なクライアント・コンピュータを示す。

【図2】並列に動作するネットワーク・ノード発見エージェントを用いてネットワークのトポロジを発見するネットワーク・ノード発見マネージャの間略化されたブロック図である。

【図3】並列に動作するネットワーク・ノード発見エージェントを用いてネットワークのトポロジを発見するネットワーク・ノード発見マネージャの動作のフローチャートである。

【図4】並列に動作するネットワーク・ノード発見エージェントを用いてネットワークのトポロジを発見するネットワーク・ノード発見マネージャの動作のフローチャートである。

【図5】並列に動作するネットワーク・ノード発見エージェントを用いてネットワークのトポロジを発見するネットワーク・ノード発見マネージャの動作のフローチャートである。

【発明を実施するための形態】

【0010】

並列に動作する複数のノード発見エージェントを用いたネットワーク・ノード発見マネージャによってネットワークのトポロジを発見する方法、システムおよびコンピュータ使用可能媒体が開示される。当業者には分かるように、本発明は、方法、システム、またはコンピュータ・プログラム製品として具現化することができる。従って、本発明の実施形態は、完全にハードウェアによって、または完全にソフトウェア(ファームウェア、常駐ソフトウェア、マイクロコードなどを含む)によって、あるいはソフトウェアおよびハードウェアを組み合わせた実施形態によって、実現することができる。こうした多様な実施形態は全て、本明細書において一般的に「回路」、「モジュール」または「システム」と称される場合もある。それに加え、本発明は、媒体に組み込まれたコンピュータ使用可能プログラム・コードを有するコンピュータ使用可能ストレージ媒体上のコンピュータ・プログラム製品の形をとることもある。

【0011】

任意の適切なコンピュータ使用可能またはコンピュータ可読媒体を利用することができる。コンピュータ使用可能またはコンピュータ可読媒体は、例えば、電子、磁気、光、電磁気、赤外線、または半導体のシステム、装置、デバイスあるいは伝搬媒体とすればよいが、これらに限定されない。コンピュータ可読媒体のさらに具体的な例(非網羅的なリスト)としては、以下を挙げることができるであろう: 1つ以上のワイヤを有する電気的接続、携帯型コンピュータ・ディスクケット、ハード・ディスク、ランダム・アクセス・メモリ(RAM: random access memory)、読み取り専用メモリ(ROM: read-only memory)、消去可能プログラム可能読み取り専用メモリ(E PROM: erasable programmable read-only memoryまたはフラッシュ・メモリ)、光ファイバ、携帯型コンパクト・ディスク読み取り専用メモリ(CD-ROM: compact disc read-only memory)、光ストレージ・デバイス、インターネットまたはイントラネットをサポートしている媒体などの伝送媒体、あるいは磁気ストレージ・デバイス。プログラムは、例えば紙またはその他の媒体の光学式走査などを介して電子的に取り込み、その後必要に応じてコンパイル、解釈、あるいは他の適切な方法で処理してからコンピュータ・メモリに保存することができるため、コンピュータ使用可能またはコンピュータ可読媒体が、プログラムが書き込まれた紙またはその他の適切な媒体である可能性すらあることに留意されたい。本文書の文脈において、コンピュータ使用可能またはコンピュータ可読媒体は、命令

10

20

30

40

50

実行システム、装置またはデバイスによってあるいはこれらに関連して使用されるように、上記のプログラムを収容、保存、通信、伝搬あるいは搬送することのできる任意の媒体とすればよい。コンピュータ使用可能媒体には、組み込まれたコンピュータ使用可能プログラム・コードを備えてベースバンドあるいは搬送波の一部として伝搬されるデータ信号を含めることもできる。コンピュータ使用可能プログラム・コードは、例えばインターネット、ワイアライン、光ファイバ・ケーブル、無線周波数 (R F : r a d i o f r e q u e n c y) などを含むがこれらに限定されない任意の適切な媒体を用いて、伝送されるとよい。

【0012】

本発明の動作を実行するコンピュータ・プログラム・コードは、Java (R)、Sm 10 a l l t a l k 、 C + + などの、オブジェクト指向プログラミング言語で書かれているとよい。一方、本発明の動作を実行するコンピュータ・プログラム・コードは同様に、「C」プログラミング言語または類似のプログラミング言語など、従来の手続き型プログラミング言語で書かれてもよい。プログラム・コードは、全面的にユーザのコンピュータ上で実行されてもよいし、自立型ソフトウェア・パッケージとして部分的にユーザのコンピュータ上で実行されてもよいし、部分的にユーザのコンピュータ上で実行され、さらに部分的にリモート・コンピュータ上で実行されてもよいし、あるいは全面的にリモート・コンピュータまたはサーバ上で実行されてもよい。後の方のシナリオでは、リモート・コンピュータが、ローカル・エリア・ネットワーク (L A N : l o c a l a r e a n e t w o r k) またはワイド・エリア・ネットワーク (W A N : w i d e a r e a n e t w o r k) を介してユーザのコンピュータに接続されてもよいし、あるいはこの接続が、外部コンピュータに対して (例えば、インターネット・サービス・プロバイダを使いインターネットを介して) 設けられてもよい。

【0013】

以下、本発明の実施形態による方法、装置 (システム) およびコンピュータ・プログラム製品のフローチャート説明図またはブロック図あるいはその両方を参照して、本発明の実施形態を説明する。当然のことながら、フローチャート説明図またはブロック図あるいはその両方の各ブロック、ならびにフローチャート説明図またはブロック図あるいはその両方の複数ブロックの組み合わせは、コンピュータ・プログラム命令によって実現することができる。こうしたコンピュータ・プログラム命令は、汎用コンピュータ、特殊用途コンピュータ、またはその他のプログラム可能なデータ処理装置のプロセッサへ提供され 30 マシンを形成するとよく、それにより、コンピュータまたは他のプログラム可能なデータ処理装置のプロセッサによって実行される命令が、フローチャートまたはブロック図あるいはその両方の単数または複数のブロックに明示されている機能 / 作用を実現する手段をもたらすようになる。

【0014】

上記のコンピュータ・プログラム命令はさらに、コンピュータまたは他のプログラム可能なデータ処理装置を特定の形で機能するように仕向けることが可能なコンピュータ可読メモリに保存されてもよく、それにより、コンピュータ可読メモリに保存された命令が、フローチャートまたはブロック図あるいはその両方の単数または複数のブロックに明示 40 されている機能 / 作用を実現する命令手段を備えた製品を形成するようになる。

【0015】

コンピュータ・プログラム命令はさらに、コンピュータまたは他のプログラム可能なデータ処理装置上にロードされて、コンピュータまたは他のプログラム可能な装置上における一連の動作ステップの実施をもたらし、コンピュータ実装されたプロセスを形成してもよく、それにより、コンピュータまたは他のプログラム可能な装置上で実行される命令が、フローチャートまたはブロック図あるいはその両方の単数または複数のブロックに明示されている機能 / 作用を実現するステップ群を提供するようになる。

【0016】

図1は、本発明が利用され得る例示的なクライアント・コンピュータ102のブロック

図である。クライアント・コンピュータ102は、システム・バス106に結合されたプロセッサ・ユニット104を含む。ディスプレイ110を制御する映像アダプタ108も、同様にシステム・バス106に結合されている。システム・バス106は、バス・ブリッジ112を介して、入力/出力(I/O: Input/Output)バス114に結合されている。I/Oインターフェース116が、I/Oバス114に結合されている。I/Oインターフェース116は、キーボード118、マウス120、コンパクト・ディスク・読み取り専用メモリ(CD-ROM)ドライブ122、フレキシブル・ディスク・ドライブ124、およびフラッシュ・ドライブ・メモリ126などの多様なI/Oデバイスに対する通信を提供する。I/Oインターフェース116に接続されているポートの形式は、コンピュータ・アーキテクチャ技術の当業者に周知の任意のものとすればよく、ユニバーサル・シリアル・バス(USB: Universal Serial Bus)ポートなどが挙げられるが、これに限定されない。
10

【0017】

クライアント・コンピュータ102は、システム・バス106に結合されているネットワーク・インターフェース130を用いネットワーク128を介して、サービス・プロバイダ・サーバ152と通信することができる。ネットワーク128は、インターネットなどの外部ネットワークとしてもよいし、あるいは、Ethernet(R)ネットワークまたは仮想プライベート・ネットワーク(VPN: Virtual Private Network)などの内部ネットワークとしてもよい。ネットワーク128を使うことでも、クライアント・コンピュータ102は、本発明を用いてサービス・プロバイダ・サーバ152にアクセスすることができる。
20

【0018】

ハード・ドライブ・インターフェース132がさらに、システム・バス106に結合されている。ハード・ドライブ・インターフェース132は、ハード・ドライブ134のインターフェースとなる。好適な一実施形態では、ハード・ドライブ134は、同様にシステム・バス106に結合されているシステム・メモリ136にポピュレートする。システム・メモリ136にポピュレートするデータとしては、クライアント・コンピュータ102のオペレーティング・システム(OS: Operating System)138およびソフトウェア・プログラム144がある。
30

【0019】

OS138は、ソフトウェア・プログラム144などの資源に対する透過的なユーザ・アクセスを提供するシェル140を含む。一般に、シェル140は、ユーザとオペレーティング・システムとの間にインターフリタおよびインターフェースを提供するプログラムである。さらに具体的に言えば、シェル140は、コマンド・ライン・ユーザ・インターフェースに入力されたあるいはファイルから得られたコマンドを、実行する。故にシェル140(UNIX(R)においてこのように呼ばれる)は、Windows(R)ではコマンド・プロセッサとも呼ばれており、一般に、オペレーティング・システムのソフトウェア階層の最上位レベルであって、コマンド・インターフリタとして機能する。シェルは、システム・プロンプトを提供し、キーボード、マウスまたは他のユーザ入力媒体により入力されたコマンドを解釈し、さらに、解釈されたコマンド(単数または複数)を、処理に向けてオペレーティング・システムの適切な下位レベル(例えば、カーネル142)へ送信する。シェル140は一般に、テキスト・ベースのライン指向のユーザ・インターフェースであるが、一方で本発明は、例えばグラフィカル、音声、ジェスチャなどその他のユーザ・インターフェース・モードをサポートすることもできる。
40

【0020】

図に示すように、OS138はさらに、OS138の下位レベルの機能を備えたカーネル142を含む。下位レベルの機能には、OS138の他の部分およびソフトウェア・プログラム144によって要求される基本的なサービスが含まれ、例えば、メモリ管理、プロセスおよびタスク管理、ディスク管理、ならびにマウスおよびキーボード管理などが挙げられる。
50

【0021】

ソフトウェア・プログラム144は、ブラウザ146と電子メール・クライアント148とを含むとよい。ブラウザ146は、ワールド・ワイド・ウェブ(WWW:World Wide Web)クライアント(すなわち、クライアント・コンピュータ102)がハイパーテキスト転送プロトコル(HTTP:HyperText Transfer Protocol)メッセージングを用いてインターネットに対しネットワーク・メッセージを送受信できるようにするプログラム・モジュールおよび命令を含み、それによりサービス・プロバイダ・サーバ152との通信を可能にしている。ソフトウェア・プログラム144は、さらに、ノード発見マネージャ150と、1つ以上のノード発見エージェント152とを含む。ノード発見マネージャ150および1つ以上のノード発見エージェント152は、本明細書で下記に示す図2乃至5に記載されるプロセスを実現するコードを含む。一実施形態では、クライアント・コンピュータ102は、ノード発見マネージャ150および1つ以上のノード発見エージェント152を、サービス・プロバイダ・サーバ152からダウンロードすることができる。10

【0022】

クライアント・コンピュータ102に図示するハードウェア要素は、網羅的であることを意図したものではなく、むしろ、本発明によって使用されるコンポーネントを強調するための代表例である。例えば、クライアント・コンピュータ102には、磁気カセット、デジタル多用途ディスク(DVD:Digital Versatile Disk)、ベルヌーイ・カートリッジなどといった、別のメモリ・ストレージ・デバイスを含めてよい。これらおよびその他の変形例は、本発明の精神および範囲のうちにあるものとされる。20

【0023】

図2は、並列に動作するネットワーク・ノード発見エージェントを用いてネットワークのトポロジを発見するネットワーク・ノード発見マネージャの間略化されたブロック図である。多様な実施形態において、ネットワーク管理ホスト202には、ネットワーク・ノード発見マネージャ150と1つ以上のネットワーク・ノード発見エージェント152とが含まれる。ネットワーク・ノード発見動作は、ネットワーク・ノード発見マネージャ150が、ターゲット・ネットワーク208の既知のネットワーク・ノードに関するノード情報を含んだNodeList204を取得することで始まる。NodeList204に含まれているノード情報には、ネットワーク208の各既知ノード「A」210、「B」212、「C」214および「D」216の、一意のノード識別子が含まれる。30

【0024】

ターゲット・ネットワーク208のトポロジ発見に使用されるノード発見エージェント152の数(「K」)が、ノード発見マネージャ150によって決められる。1つのノード発見エージェントから既知ノードの総数に至るまで、任意の数のノード発見エージェントを選択することができる(すなわち、1~Kノードの総数)。当然のことながら、1つのノード発見エージェントでは、何千ものノードを含むネットワークのトポロジを発見するには不適当な場合がある。反対に、何千ものノードの各々について別個のノード発見エージェントを用いれば、恐らく、ネットワーク管理ホスト202に過度の処理オーバーヘッドを課すことになるであろう。既知ノードおよび発見されたノードの双方に関するノード情報を保持するために、ネットワーク・ノードのGraphList206がノード発見マネージャ150によって作成され、最初は、空に指定される。40

【0025】

既知ノード「A」210、「B」212、「C」214および「D」216を、ノード発見エージェント152の数(「K」)と同数のノード・グループに分割するために、NodeList204内の1意のノード識別子にハッシュ演算が施される。例として、既知ノード「A」210および「B」212の1意のノード識別子に施されるハッシュ演算は、ノード・グループ「1」222に割り当てられたグループ・ハッシュ値を生じさせる。同様に、既知ノード「C」214および「D」216の1意のノード識別子に施される50

ハッシュ演算は、ノード・グループ「2」224に割り当てられたノード・グループ・ハッシュ値を生じさせる。ノード・グループ「1」222および「2」224に関連している生成ハッシュ値は、その後、NodeList204内の対応する一意のノード識別子に付け加えられる。ノード・グループ「1」222および「2」224に関連しているハッシュ値は、次に、ノード発見エージェント「M」152にそれぞれ割り当てられる。このとき、 $1 \leq M \leq K$ である。ノード・グループ・ハッシュ値が各ノード発見エージェント「M」152にそれぞれ割り当てられたら、ネットワーク・ノード発見マネージャ150によって、「K」個のノード発見エージェント152の並列動作が開始される。ノード発見マネージャ150は、GraphList206のノード情報受け取りスレッドを開始し、続いてノード発見エージェント「M」152各自のノード発見スレッドを開始する。

10

【0026】

その後、未処理のノード識別子がNodeList204からポップされ、対応するノード・グループ・ハッシュ値が特定される。当該ノード・グループ・ハッシュ値に割り当てられているノード発見エージェント「M」152がビジーであると判断された場合には、ビジーでないノード発見エージェント「M」152に対応するノード識別子エントリがポップされるまで、上記のプロセスが繰り返される。ビジーでないノード発見エージェント「M」152に対応するノード・グループ・ハッシュ値を有するノード識別子エントリがポップされたら、ターゲットの既知ノード「A」210、「B」212、「C」214、「D」216を、処理に向けて、そのノード識別子エントリの対応するノード発見エージェント「M」152に割り当てる。その後、ノード発見エージェント「M」152は、SNMPクエリおよび当業者にはよく知られたその他の方式といったプローブ・メッセージを、ターゲットの既知ノード「A」210、「B」212、「C」214および「D」216へ送信する。

20

【0027】

次に、既知のターゲット・ノード「A」210、「B」212、「C」214および「D」216が1つ以上の未知の周辺ノード「E」218、「F」220に接続されているかどうかの判断が下される。ターゲット・ノード「A」210、「B」212、「C」214、「D」216が周辺ノード「E」218、「F」220に接続されていると判断された場合、プローブは、ターゲット・ノード「A」210、「B」212、「C」214、「D」216から、周辺ノード「E」218、「F」220との接続に関する情報を含んだノード情報を収集する。一実施形態では、接続は物理的接続である。別の実施形態では、接続は論理的である。さらに別の実施形態では、接続は物理的もしくは論理的接続の組み合わせであってもよい。収集されたノード情報を処理されて、発見された周辺ノード「E」218および「F」220の一意のノード識別子が特定される。

30

【0028】

一例として、ノード・グループ「2」224に割り当てられたノード発見エージェント「M」152が、周辺ノードとの接続に関して既知ノード「D」216を調べる。収集されたノード情報は、既知ノード「D」216が既知ノード「A」210、「B」212、「C」214のほか、未知の、すなわち発見されたノード「E」218および「F」220に接続されていることを示している。そこで、発見された周辺ノード「E」218および「F」220の一意のノード識別子がNodeList204に現在記載されているかどうかの判断が下される。記載されていない場合、発見された周辺ノードそれぞれの一意のノード識別子に、本明細書にて詳述されるハッシュ演算が施されてハッシュ値が生成され、このハッシュ値を用いることで、発見された周辺ノード「E」218および「F」220を、処理に向けて、指定されているノード発見エージェント「M」152へ割り当てることができる。一例として、発見された周辺ノード「E」218のハッシュ値は、この周辺ノードがノード・グループ「1」222に割り当てられること226を示す。同様に、発見された周辺ノード「F」220のハッシュ値は、この周辺ノードがノード・グループ「2」224に割り当てられること228を示す。その後、ターゲット・ノード「A」210、「B」212、「C」214、「D」216と、発見された周辺ノード「E」2

40

50

18、「F」220との1つ以上の接続が、GraphList206に記載されているかどうかの判断が下される。記載されていない場合、これらの接続がGraphList206に追加される。上記のプロセスは、「K」個のノード発見エージェント「M」152全てが、既知ノード「A」210、「B」212、「C」214、「D」216ならびに発見された周辺ノード「E」218、「F」220に関する各自のノード発見動作を完了するまで、継続される。全てのノード情報がGraphList206に記載されたら、ネットワーク・トポロジが生成され、ネットワーク・ノード発見動作は終了する。

【0029】

図3乃至5は、並列に動作するネットワーク・ノード発見エージェントを用いてネットワークのトポロジを発見するネットワーク・ノード発見マネージャのフローチャートである。本発明の一実施形態では、ステップ302にて、ネットワーク・ノード発見動作が始まる。ステップ304にて、ネットワーク・ノード発見マネージャがターゲット・ネットワークに関する情報を取得し、この情報には、当該ネットワーク内の既知ノード各々の一意のノード識別子が含まれている。次にステップ306にて、ターゲット・ネットワークのトポロジ発見に用いられるノード発見エージェントの数（「K」）が、ノード発見マネージャによって決められる。例えば、1つのノード発見エージェントから既知ノードの総数に至るまで、任意の数のノード発見エージェントを選択することができる（すなわち、1 K ノードの総数）。当然のことながら、1つのノード発見エージェントでは、何千ものノードを含むネットワークのトポロジを発見するには不十分な場合がある。反対に、何千ものノード各々について個別のノード発見エージェントを用いれば、恐らく、ネットワーク管理サーバに過度の処理オーバーヘッドを課すことになるであろう。

【0030】

次にステップ308にて、ノード発見マネージャは、既知ノードの最初のセットを、関連している一意のノード識別子とともに、NodeListに割り当てる。ステップ310にて、既知ノードおよび発見されたノードの双方に関するノード情報を保持することになるGraphListが、ノード発見マネージャによって作成され、最初は、空に指定される。その後ステップ312にて、既知ノード群をノード発見エージェントの数（「K」）と同数のノード・グループに分割するために、NodeList内の一意のノード識別子にハッシュ演算が施される。次いでステップ314にて、生成ハッシュ値の各々が、NodeListにおいて一意のノード識別子にそれぞれ関連付けられる。各ノード・グループに関連しているハッシュ値は、その後ステップ316にて、ノード発見エージェント「M」に割り当てられるが、この場合に、1 M Kである。グループ・ハッシュ値が各ノード発見エージェント「M」に割り当てられたら、ステップ318にて、「K」個のノード発見エージェントの並列動作が開始される。ノード発見マネージャによって、ステップ320にて、GraphListのノード情報受け取りスレッドが開始され、続いてステップ322にて、ノード発見エージェント「M」各々のノード発見スレッドが開始される。

【0031】

その後ステップ324にて、NodeListにある全てのエントリが処理済みであるかどうかの判断が下される。全てのエントリが処理済みであれば、ステップ326にて、「K」個のノード発見エージェント全てがトポロジ発見動作を完了したかどうかの判断が下される。全てのエントリが処理済みでなければ、ステップ328にて、未処理のノード識別子がNodeListからポップされ、対応するノード・グループ・ハッシュ値が特定される。その後ステップ330にて、当該ノード・グループ・ハッシュ値に割り当てられているノード発見エージェント「M」がビジーであるかどうかの判断が下される。ビジーである場合には、ビジーでないノード発見エージェント「M」に対応するノード識別子エントリがポップされるまで、上記のプロセスをステップ328から始めて繰り返す。ポップされたノード識別子エントリが、ビジーでないノード発見エージェント「M」に対応するノード・グループ・ハッシュ値を有することがステップ330にて判断されたら、ターゲット・ノードを、処理に向けてノード発見エージェント「M」に割り当てる。その

10

20

30

40

50

後ステップ334にて、ノード発見エージェント「M」が、SNMPクエリおよび当業者にはよく知られたその他の方式といったプローブ・メッセージを、ターゲット・ノードへ送信する。

【0032】

次にステップ336にて、ターゲット・ノードが1つ以上の周辺ノードに接続されているかどうかの判断が下される。ステップ336にて当該ノードはネットワークの端点であると判断された場合、あるいは当該ノードは他のいずれのノードにも接続されていないと判断された場合には、そのノード識別子は、ステップ364にて、処理済みノードとしてNodeList内でマークされる。その後、上記のプロセスを、NodeListにある全エントリが処理済みであるかどうかの判断が下されるステップ324から始めて繰り返す。一方、ステップ336にてターゲット・ノードは周辺ノードに接続されていると判断された場合、プローブは、当該ノードから、周辺ノードとの接続に関する情報を含んだノード情報を収集する。一実施形態では、接続は物理的接続である。別の実施形態では、接続は論理的である。さらに別の実施形態では、接続は物理的もしくは論理的接続の組み合わせであってもよい。次にステップ340にて、収集されたノード情報がNodeConnectionリストに変換され、次いでこのNodeConnectionリストが、ノード発見マネージャへ戻される。

【0033】

次にステップ342にて、NodeConnectionリストにある全てのノード接続エントリが処理済みであるかどうかの判断が下される。ステップ342にて全てのノード接続エントリが処理済みではないと判断されたら、ステップ344にて、周辺ノード・エントリがポップされて、その一意のノード識別子がステップ344にて特定される。次にステップ346にて、上記の一意のノード識別子がNodeListに現在記載されているかどうかの判断が下される。記載されている場合には、その後ステップ348にて、その周辺ノードが処理済みであるかどうかの判断が下される。

【0034】

NodeListにてエントリとして表される、発見された周辺ノードは、対応するノード・グループ・ハッシュ値を有することになり、このノード・グループ・ハッシュ値によって、所定のノード発見エージェント「M」による処理に向けて割り当てられる事になるのは、当業者には明らかであろう。ノード発見エージェント「M」がカレント・エージェントであるかその他のエージェントであるかにかかわらず、上記のノード・エントリは、まだステップ328でポップされていない、すなわちステップ332でノード発見エージェント「M」に割り当てられていないものである。従って、更なる処理動作は、ひとたびこの周辺ノード・エントリがポップされればステップ328から始まって実施されることになるので、本プロセスのこの時点では必要ない。故に、ステップ348にてその周辺ノードが処理済みではないと判断されたら、上記のプロセスをステップ342から始めて繰り返す。

【0035】

一方、ステップ348にて、その周辺ノード・エントリがNodeListにおいて処理済みとしてマークされていると判断された場合には、ステップ350にて、ターゲット・ノードと周辺ノードとの1つ以上の接続が、物理的であれ論理的であれ、GraphListに記載されているかどうかの判断が下される。当然のことながら、たとえ、ターゲット・ノードおよび周辺ノードの双方が以前に処理されていたとしても、それらが処理された時以降に新たな接続が構築されることはある。そのような新たな接続は、ネットワークのトポロジを変えてしまい、その精度に悪影響を与えかねない。従って、ステップ350にて、ターゲット・ノードと周辺ノードとの1つ以上の接続がGraphListに記載されていないと判断された場合には、それらをステップ352にてGraphListに追加する。その他の場合には、上記のプロセスをステップ342から始めて繰り返す。

【0036】

10

20

30

40

50

ステップ346にて周辺ノードの一意のノード識別子がNodeListに現在記載されていないと判断された場合、ステップ354にて、その一意のノード識別子にハッシュ演算が施される。本明細書にて詳細に記載するように、ハッシュ演算は、発見された周辺ノードを既存のノード・グループに関連付けるのに用いられるハッシュ値を生成する。ひいては、ノード・グループ・ハッシュ値は、発見された周辺ノードを、処理に向けて所定のノード発見エージェント「M」へ割り当てることにもなることが分かる。そのため、ステップ356にて、ハッシュ値を周辺ノードの一意のノード識別子に追加し、これを後続処理に向けてエントリとしてNodeListに加える。

【0037】

本明細書にて上述したように、発見された周辺ノードをNodeListにおいて表す、対応するノード・グループ・ハッシュ値は、発見された周辺ノードを所定のノード発見エージェント「M」による処理に向けて割り当てる結果をもたらす。そのため、更なるノード接続処理動作は、ひとたび周辺ノード・エントリがポップされればステップ328から始まって実施されることになるので、本プロセスのこの時点では必要ない。そこで、上記のプロセスを、ステップ342から始めて繰り返す。ステップ326にて全てのノード発見エージェント「M」がそれぞれのノード発見動作を完了したと判断されたら、ステップ358にて、GraphyDiscoveryリストに含まれているノード情報からネットワーク・トポロジが生成される。ステップ358にてネットワーク・トポロジが生成されたら、ステップ360にて、ネットワーク・ノード発見動作が終了する。

【0038】

図面のフローチャートおよびブロック図は、本発明の多様な実施形態によるシステム、方法およびコンピュータ・プログラム製品の、あり得る実装のアーキテクチャ、機能、および動作を示すものである。この点に関連し、フローチャートまたはブロック図の各ブロックが、明示された論理機能（単数または複数）を実現する1つ以上の実行可能命令を含んだモジュール、セグメント、またはコードの一部を表すこともある。さらに、一部の代替的な実装では、ブロックに示された機能が、図面に示された順序から外れて生じ得ることにも留意する必要がある。例えば、関与する機能次第で、連続して示された2つのブロックがほぼ同時に実行されることもあり、時にはそれらのブロックが逆の順序で実行されることもある。さらに、ブロック図またはフローチャート説明図あるいはその両方の各ブロック、ならびにブロック図またはフローチャート説明図あるいはその両方の複数ブロックの組み合わせは、明示された機能または作用を実行する特殊用途向けハードウェアベースのシステムによって、あるいは特殊用途向けハードウェアとコンピュータ命令との組み合わせによって実現することが可能であることも分かる。

【0039】

本明細書にて使用される専門用語は、特定の実施形態を説明することを目的としているものに過ぎず、本発明を限定しようとするものではない。本明細書で使用される「1つの（a）、（an）」および「その（the）」という単数形は、そうではないことが文脈によって明確に指示されない限り、複数形をも同様に含めようとするものである。さらに当然のことながら、用語「含む」または「含んでいる」あるいはその両方は、本明細書での使用に当たり、提示される特徴、整数値、ステップ、動作、要素またはコンポーネントあるいはその全ての存在を明示するものであるが、1つ以上のその他の特徴、整数値、ステップ、動作、要素、コンポーネントまたはこれらのグループあるいはその全ての存在または追加を排除するものではない。

【0040】

以下の請求項における全てのミーンズまたはステップ・プラス・ファンクション要素の対応する構造、材料、作用および同等物は、請求されているその他の要素と組み合わされて特に請求されているように当該機能を果たすあらゆる構造、材料または行為を包含しようとするものである。説明および記述を目的として本発明に関する記載を提示してきたが、網羅的であること、または開示された形態に本発明を限定することは意図されていない。多数の変更および変形が、本発明の範囲および精神から逸脱することなく、当業者に明

10

20

30

40

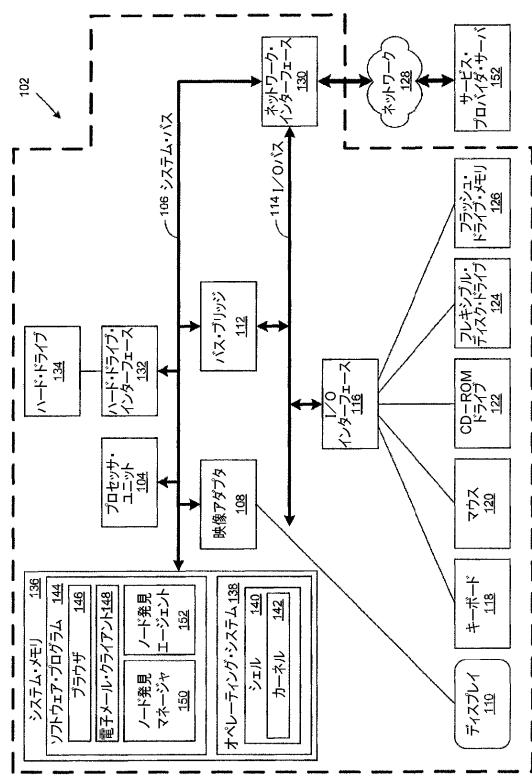
50

らかとなるであろう。実施形態は、本発明の原理および実際的応用を最もよく説明するために、加えて他の当業者が、検討される個々の用途に適するように多様な変更を伴った多様な実施形態に向けて本発明を理解できるようにするために、選択され記載されたものである。

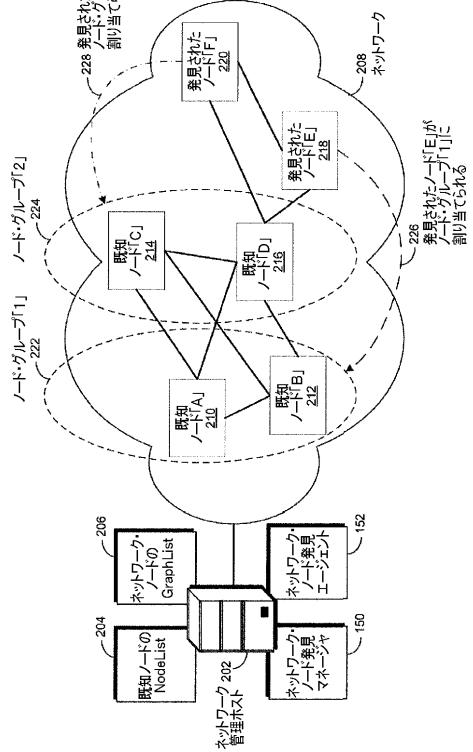
【 0 0 4 1 】

このように、本願の発明を詳細に、発明の好適な実施形態に関連して記載してきたが、当然のことながら、添付の請求項にて定義される本発明の範囲を逸脱することなく、変更および変形が実現可能である。

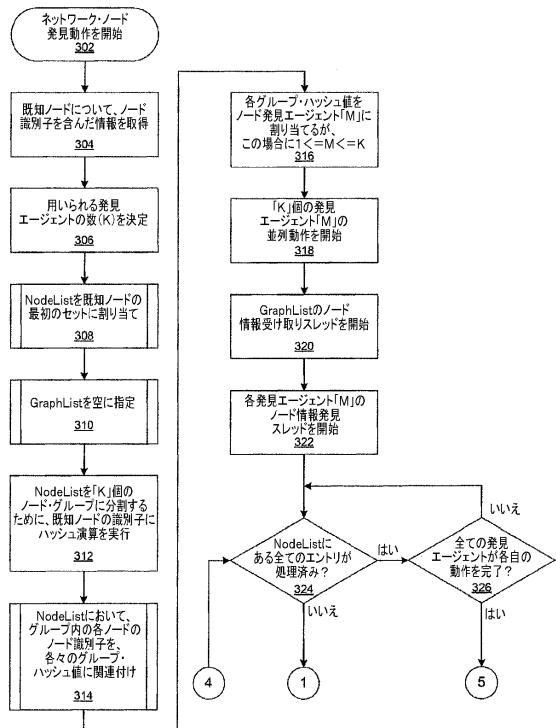
【 図 1 】



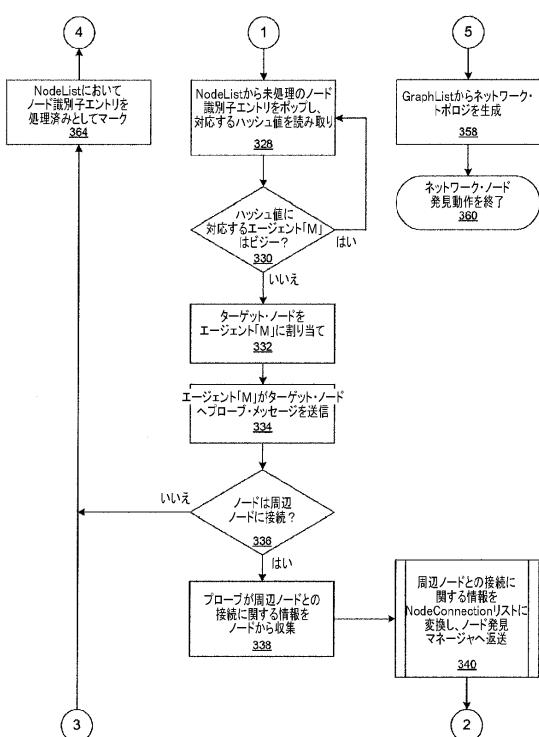
【 図 2 】



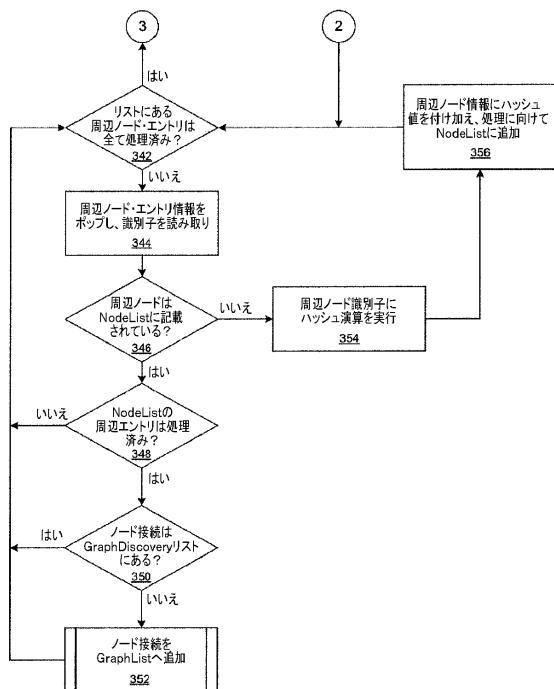
【図3】



【図4】



【図5】



フロントページの続き

(72)発明者 ヴェルマ、ディネシュ

アメリカ合衆国10549 ニューヨーク州 ニュー・キャッスル キスコ・パーク・ドライブ
56

(72)発明者 マーチン、ダニエル、ジョセフ

アメリカ合衆国12564 ニューヨーク州 ポーリング テン・ハンター・レーン

審査官 永井 啓司

(56)参考文献 特開平04-229742 (JP, A)

特開2004-282272 (JP, A)

特開平09-093265 (JP, A)

(58)調査した分野(Int.Cl., DB名)

H04L 12/00 - 12/28, 12/44 - 12/955