

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4277918号
(P4277918)

(45) 発行日 平成21年6月10日(2009.6.10)

(24) 登録日 平成21年3月19日(2009.3.19)

(51) Int.Cl.

F I

G O 6 F 13/00 (2006.01)

G O 6 F 17/30 (2006.01)

G O 6 F 13/00 3 5 1 A

G O 6 F 17/30 3 4 O Z

G O 6 F 17/30 4 1 4 B

G O 6 F 17/30 1 1 O C

請求項の数 18 (全 44 頁)

(21) 出願番号 特願2007-106446 (P2007-106446)
 (22) 出願日 平成19年4月13日(2007.4.13)
 (65) 公開番号 特開2008-262507 (P2008-262507A)
 (43) 公開日 平成20年10月30日(2008.10.30)
 審査請求日 平成20年6月11日(2008.6.11)

(出願人による申告)平成18年度、総務省、「ユビキ
 タスネットワーク制御・管理技術の研究開発」委託研究
 、産業再生法第30条の適用を受ける特許出願

早期審査対象出願

(73) 特許権者 000004237
 日本電気株式会社
 東京都港区芝五丁目7番1号
 (74) 代理人 100103090
 弁理士 岩壁 冬樹
 (74) 代理人 100124501
 弁理士 塩川 誠人
 (72) 発明者 中台 慎二
 東京都港区芝五丁目7番1号 日本電気株
 式会社内

審査官 千本 潤介

最終頁に続く

(54) 【発明の名称】 データ検索装置、データ検索方法およびデータ検索用プログラム

(57) 【特許請求の範囲】

【請求項1】

ネットワーク上に配置され、他の装置と共有するデータを検索するデータ検索装置であ
 って、

他のデータ検索装置と区別するために付与される識別子であって、複数ビットで構成さ
 れた論理識別子が付与され、

検索対象となる各データ検索装置の論理識別子の集合である論理識別子範囲を指定した
 検索要求を受け取ると、前記論理識別子範囲に含まれる各論理識別子のうちの最大値であ
 る最大論理識別子と最小値である最小論理識別子とに基づいて、指定された前記論理識別
 子範囲を分割する分割処理を実行する分割手段と、

前記分割手段によって分割された各論理識別子範囲を検索対象として指定した検索要求
 を、転送先の装置に転送する検索要求転送手段と、

該データ検索装置と、該データ検索装置の論理識別子に対して時計回りの次の論理識
 別子に対応する装置である Successor との間に位置するノードから任意に選択し
 た仮想ノードの識別子である仮想識別子を生成する検索実行手段と、

前記分割手段は、前記最大論理識別子と前記最小論理識別子に含まれる先頭ビットが同
 一である場合に、指定された前記論理識別子範囲に含まれる各論理識別子を先頭ビットを
 押し出すようにビットシフトしたものを1つの論理識別子範囲とする処理を実行し、前記
 最大論理識別子と前記最小論理識別子に含まれる先頭ビットが異なる場合に、指定された
 前記論理識別子範囲における各論理識別子の先頭ビットを押し出すようにビットシフトし

10

20

たものの集合についてシフト後の前記最小論理識別子以上の各論理識別子を含む論理識別子範囲とシフト後の前記最大論理識別子以下の各論理識別子を含む論理識別子範囲とを得るビットシフト範囲分割処理を実行し、

前記分割手段は、

指定された前記論理識別子範囲を前記 S u c c e s s o r の論理識別子を境界にして 2 つに分割する範囲分割処理を実行する範囲分割手段を含み、

該データ検索装置の論理識別子が表す数値と前記 S u c c e s s o r の論理識別子が表す数値とで示される範囲である特定範囲に前記仮想ノードの論理識別子が含まれる場合に、前記範囲分割手段によって分割された各論理識別子範囲のうち論理識別子が表す数値が大きい方の論理識別子範囲に対して前記ビットシフト範囲分割処理を実行し、

前記検索要求転送手段は、前記範囲分割手段によって分割された各論理識別子範囲を検索対象として指定するとともに前記仮想ノードの論理識別子を含む検索要求を、論理識別子範囲に含まれる装置に転送する

データ検索装置。

【請求項 2】

前記分割手段は、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子が前記特定範囲に含まれる場合に、前記範囲分割処理および前記ビットシフト範囲分割処理を実行する

請求項 1 記載のデータ検索装置。

【請求項 3】

前記分割手段は、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子が前記特定範囲に含まれない場合、前記検索要求を前記検索要求転送手段に送る

請求項 1 または 2 記載のデータ検索装置。

【請求項 4】

前記検索実行手段は、仮想ノードの論理識別子に含まれる後尾のビットが、前記受け取った検索要求で指定される論理識別子範囲に含まれる各論理識別子の特定の位置のビットとなるように前記仮想ノードの論理識別子を生成する

請求項 1 記載のデータ検索装置。

【請求項 5】

前記データ検索装置は、

前記分割手段によって分割された論理識別子範囲毎に、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子と該論理識別子範囲に含まれる各論理識別子の先頭のビットとから仮想ノードの論理識別子を算出する先頭ビット取得手段を備えた

請求項 1 または 2 記載のデータ検索装置。

【請求項 6】

前記分割手段は、

前記範囲分割処理によって分割された論理識別子範囲のうち前記特定範囲に含まれない論理識別子範囲を、あらかじめ設定された仮想範囲をもとに分割する第 2 分割手段と、

前記第 2 分割手段によって分割された論理識別子範囲のうち前記特定範囲に含まれる論理識別子範囲を左ビットシフトするとともに、シフト回数を増加させる範囲左ビットシフト手段とを含み、

前記検索要求転送手段は、検索対象の論理識別子範囲とともに仮想範囲とシフト回数を含む検索要求を転送する

請求項 1 記載のデータ検索装置。

【請求項 7】

ネットワーク上に配置され、他の装置と共有するデータを検索するデータ検索装置で実行されるデータ検索方法であって、

前記データ検索装置には、他のデータ検索装置と区別するために付与される識別子であ

10

20

30

40

50

って、複数ビットで構成された論理識別子が付与され、

検索対象となる各データ検索装置の論理識別子の集合である論理識別子範囲を指定した検索要求を受け取ると、前記論理識別子範囲に含まれる各論理識別子のうちの最大値である最大論理識別子と最小値である最小論理識別子とに基づいて、指定された前記論理識別子範囲を分割する分割ステップと、

前記分割ステップによって分割された各論理識別子範囲を検索対象として指定した検索要求を、転送先の装置に転送する検索要求転送ステップと、

該データ検索装置と、該データ検索装置の論理識別子に対して時計回りの次の論理識別子に対応する装置である Successor との間に位置するノードから任意に選択した仮想ノードの識別子である仮想識別子を生成する検索実行ステップとを実行し、

10

前記分割ステップで、前記最大論理識別子と前記最小論理識別子に含まれる先頭ビットが同一である場合に、指定された前記論理識別子範囲に含まれる各論理識別子を先頭ビットを押し出すようにビットシフトしたものを1つの論理識別子範囲とする処理を実行し、前記最大論理識別子と前記最小論理識別子に含まれる先頭ビットが異なる場合に、指定された前記論理識別子範囲における各論理識別子の先頭ビットを押し出すようにビットシフトしたものの集合についてシフト後の前記最小論理識別子以上の各論理識別子を含む論理識別子範囲とシフト後の前記最大論理識別子以下の各論理識別子を含む論理識別子範囲とを得るビットシフト範囲分割ステップを実行し、

前記分割ステップは、指定された前記論理識別子範囲を前記 Successor の論理識別子を境界にして2つに分割する範囲分割ステップを含み、

20

前記分割ステップで、該データ検索装置の論理識別子が表す数値と前記 Successor の論理識別子が表す数値とで示される範囲である特定範囲に前記仮想ノードの論理識別子が含まれる場合に、前記範囲分割ステップによって分割された各論理識別子範囲のうち論理識別子が表す数値が大きい方の論理識別子範囲に対して前記ビットシフト範囲分割ステップを実行し、

前記検索要求転送ステップで、前記範囲分割ステップによって分割された各論理識別子範囲を検索対象として指定するとともに前記仮想ノードの論理識別子を含む検索要求を、論理識別子範囲に含まれる装置に転送する

データ検索方法。

【請求項8】

30

前記分割ステップで、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子が前記特定範囲に含まれる場合に、前記範囲分割ステップおよび前記ビットシフト範囲分割ステップを実行する

請求項7記載のデータ検索方法。

【請求項9】

前記分割ステップで、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子が前記特定範囲に含まれない場合、前記検索要求を前記検索要求転送ステップに送る

請求項7または請求項8記載のデータ検索方法。

【請求項10】

40

前記検索実行ステップで、仮想ノードの論理識別子に含まれる後尾のビットが、前記受け取った検索要求で指定される論理識別子範囲に含まれる各論理識別子の特定の位置のビットとなるように前記仮想ノードの論理識別子を生成する

請求項7記載のデータ検索方法。

【請求項11】

前記データ検索装置は、

前記分割ステップによって分割された論理識別子範囲毎に、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子と該論理識別子範囲に含まれる各論理識別子の先頭のビットとから仮想ノードの論理識別子を算出する

請求項8記載のデータ検索方法。

50

【請求項 1 2】

前記分割ステップは、

前記範囲分割ステップによって分割された論理識別子範囲のうち前記特定範囲に含まれない論理識別子範囲を、あらかじめ設定された仮想範囲をもとに分割する第 2 分割ステップと、

前記第 2 分割ステップによって分割された論理識別子範囲のうち前記特定範囲に含まれる論理識別子範囲を左ビットシフトするとともに、シフト回数を増加させる範囲左ビットシフトステップとを含み、

前記検索要求転送ステップで、検索対象の論理識別子範囲とともに仮想範囲とシフト回数を含む検索要求を転送する

請求項 7 記載のデータ検索方法。

【請求項 1 3】

ネットワーク上に配置されたデータ検索装置が他の装置と共有するデータを検索するためのデータ検索用プログラムであって、

前記データ検索装置には、他のデータ検索装置と区別するために付与される識別子であって、複数ビットで構成された論理識別子が付与され、

前記データ検索装置に、

検索対象となる各データ検索装置の論理識別子の集合である論理識別子範囲を指定した検索要求を受け取ると、前記論理識別子範囲に含まれる各論理識別子のうちの最大値である最大論理識別子と最小値である最小論理識別子とに基づいて、指定された前記論理識別子範囲を分割する分割処理と、

前記分割処理によって分割された各論理識別子範囲を検索対象として指定した検索要求を、転送先の装置に転送する検索要求転送処理と、

該データ検索装置と、該データ検索装置の論理識別子に対して時計回りの次の論理識別子に対応する装置である Successor との間に位置するノードから任意に選択した仮想ノードの識別子である仮想識別子を生成する検索実行処理とを実行させ、

前記分割処理で、前記最大論理識別子と前記最小論理識別子に含まれる先頭ビットが同一である場合に、指定された前記論理識別子範囲に含まれる各論理識別子を先頭ビットを押し出すようにビットシフトしたものを 1 つの論理識別子範囲とする処理を実行し、前記最大論理識別子と前記最小論理識別子に含まれる先頭ビットが異なる場合に、指定された前記論理識別子範囲における各論理識別子の先頭ビットを押し出すようにビットシフトしたものの集合についてシフト後の前記最小論理識別子以上の各論理識別子を含む論理識別子範囲とシフト後の前記最大論理識別子以下の各論理識別子を含む論理識別子範囲とを得るビットシフト範囲分割処理を実行させ、

前記分割処理は、指定された前記論理識別子範囲を前記 Successor の論理識別子を境界にして 2 つに分割する範囲分割処理を含み、

前記分割処理で、該データ検索装置の論理識別子が表す数値と前記 Successor の論理識別子が表す数値とで示される範囲である特定範囲に前記仮想ノードの論理識別子が含まれる場合に、前記範囲分割処理によって分割された各論理識別子範囲のうち論理識別子が表す数値が大きい方の論理識別子範囲に対して前記ビットシフト範囲分割処理を実行させ、

前記検索要求転送処理で、前記範囲分割処理によって分割された各論理識別子範囲を検索対象として指定するとともに前記仮想ノードの論理識別子を含む検索要求を、論理識別子範囲に含まれる装置に転送させる

データ検索用プログラム。

【請求項 1 4】

前記データ検索装置に、

前記分割処理で、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子が前記特定範囲に含まれる場合に、前記範囲分割処理および前記ビットシフト範囲分割処理を実行させる

10

20

30

40

50

請求項 1 3 記載のデータ検索用プログラム。

【請求項 1 5】

前記データ検索装置に、

前記分割処理で、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子が前記特定範囲に含まれない場合、前記検索要求を前記検索要求転送処理に送らせる

請求項 1 3 または請求項 1 4 記載のデータ検索用プログラム。

【請求項 1 6】

前記データ検索装置に、

前記検索実行処理で、仮想ノードの論理識別子に含まれる後尾のビットが、前記受け取った検索要求で指定される論理識別子範囲に含まれる各論理識別子の特定の位置のビットとなるように前記仮想ノードの論理識別子を生成させる

請求項 1 3 記載のデータ検索用プログラム。

【請求項 1 7】

前記データ検索装置に、

前記分割処理によって分割された論理識別子範囲毎に、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子と該論理識別子範囲に含まれる各論理識別子の先頭のビットとから仮想ノードの論理識別子を算出させる

請求項 1 5 記載のデータ検索用プログラム。

【請求項 1 8】

前記データ検索装置に、

前記分割処理で、

前記範囲分割処理によって分割された論理識別子範囲のうち前記特定範囲に含まれない論理識別子範囲を、あらかじめ設定された仮想範囲をもとに分割する第 2 分割処理と、

前記第 2 分割処理によって分割された論理識別子範囲のうち前記特定範囲に含まれる論理識別子範囲を左ビットシフトするとともに、シフト回数を増加させる範囲左ビットシフト処理とを実行させ、

前記検索要求転送処理で、検索対象の論理識別子範囲とともに仮想範囲とシフト回数を含む検索要求を転送させる

請求項 1 3 記載のデータ検索用プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ネットワークに配置された各端末で共有するデータを検索するデータ検索装置、データ検索方法およびデータ検索用プログラムに関し、特に次数が定数であるグラフを用いることを特徴としたデータ検索装置、データ検索方法およびデータ検索用プログラムに関する。

【背景技術】

【0002】

オーバーレイネットワークは、既存のリンクを用いて、その上位層における目的に応じて仮想的なリンクを形成し構成するネットワークである。すなわち、コンピュータネットワークの下位層のトポロジとは関係なく構築された上位層のネットワークをオーバーレイ・ネットワークと呼ぶ。例えば、IP ネットワークのトポロジとは無関係に構築された P2P (Peer to Peer) ネットワークなどを指す。

【0003】

オーバーレイネットワークでは、中央サーバを必要とせず、容易に柔軟な耐故障性に優れたネットワークを構築できる。ノード（ピアともいう）は自由に参加・離脱を行い、ネットワークトポロジは動的に変化する。全てのノードに対等の権限が与えられ、ネットワークに接続されている全てのノードが持つデータにアクセスすることが可能となる。

【0004】

10

20

30

40

50

オーバレイネットワークにおいて、各ノードが隣接するノードの情報（ＩＰアドレス等）のみを持つ場合は、データを検索する方法として、隣接するノードに問合せ（クエリ）をリレーで渡していく方法が採られることになる。このような検索方法の場合、検索用のネットワークポロジを維持する必要がないためアドホック性に優れ、複製を利用することで耐故障性も優れているが、検索に時間がかかるためスケーラビリティに欠ける。一方、各ノードがネットワーク上の全てのノードの情報を持っている場合は、検索時間はかかるがスケーラビリティとアドホック性の点で適切でない。このように、オーバレイネットワーク（Ｐ２Ｐネットワーク）では、要求されたデータを持つノードをどのように検索するかが重要と問題となる。

【０００５】

10

オーバレイネットワーク上の各ノードに経路情報として分散ハッシュテーブル（Distributed Hash Table：DHT）を持たせることにより、膨大なデータを高速（効率的）に検索する手法が用いられている。分散ハッシュテーブルでは、データはハッシュ空間上に配置され、各ノードがある値域を担当して値域内に存在するデータを保持、管理する。所定のノードがキーを通じてデータの検索を行う場合、キーをハッシュ関でハッシュ値に変換してハッシュ空間にマッピングを行い、目的のノードとデータを取得する。

【０００６】

分散ハッシュテーブルを用いた検索アルゴリズム（ルーティングアルゴリズム）として複数のものが考えられているが、その中にChordとKoordeが存在する。Chordは、環状のハッシュ空間上でフィンガーテーブル（Finger Table；自身のノードのIDの1, 2, 4, 8, ... 先のIDのノードの情報が登録されたテーブル）にもとづいて二分検索に似た検索を行うことでホップ数 $\log(N)$ の検索効率を実現したアルゴリズムである。また、Koordeは、Chordのフィンガーテーブルの代わりにde Bruijnグラフを利用してルーティングを行うアルゴリズムである。非特許文献1には、Koordeによるデータ検索方法が記載されている。以下、非特許文献1に記載された従来のデータ検索方法について説明する。

20

【０００７】

（１）第１の従来技術について：

図１７は、第１の従来技術のデータ検索手順を示すフローチャートである。また、図１８は、de Bruijnグラフを示す説明図である。第１の従来技術では、図１７に示すフローチャートに記載した手順に従って、論理識別子Keyが与えられたデータを、図１８に示すグラフ内のどのノードが管理するかを決定する。

30

【０００８】

図１８はde Bruijnグラフと呼ばれる。de Bruijnグラフ $D_k(n)$ は、集合 $Z_k = \{0, 1, \dots, k-1\}$ の要素からなるビット長 n のビット列 x が、全てのあるビット列を左ビットシフトして、最下位ビットに $x_0 \dots Z_k$ を加えて得られる全てのビット列に対して弧を持つグラフである。図１８に示す例では、 $k=2$ 、 $n=4$ としている。

【０００９】

40

すなわち、de Bruijnグラフでは、 n ビットの論理識別子（ノードID）を持つノードが 2^n 個存在した場合、各々のノード m から $(2m \bmod 2^n)$ 、 $(2m+1 \bmod 2^n)$ のノードにリンクが張られる。例えば、3ビットの論理識別子が xyz のノード（ID= xyz のノード）からは、論理識別子が $yz0$ のノード（ID= $yz0$ のノード）と論理識別子が $yz1$ のノード（ID= $yz1$ のノード）に対して一方向のリンクが張られる。

【００１０】

Koordeでは、このようなde Bruijnグラフを一方向リンクを利用してルーティングを行う。例えば、3ビットの論理識別子が xyz のノードから3ビットの論理識別子が abc のノードにメッセージを転送するときは、論理識別子が xyz $yz a$

50

z a b a b c のノードへ順にメッセージがルーティングされる。すなわち、終点の論理識別子と始点の論理識別子をつなげた x y z a b c を順に左にシフトさせたノードにメッセージが転送される。

【 0 0 1 1 】

データの論理識別子を管理するノードを探す例として、論理識別子が 1 3 (2 進数表示で 1 1 0 1) である図 1 8 に示すノード 2 1 3 が、論理識別子が 9 (2 進数表示で 1 0 0 1) であるデータを格納するノードを検索する場合について説明する。

【 0 0 1 2 】

図 1 7 に示す処理において、ノード 2 1 3 が、論理識別子 K e y の値「 1 0 0 1 」を、キーシフト k s h i f t で探す。初期キーシフトの値は、データの論理識別子 K e y の値と同一とする。すなわち、初期キーシフトの値を「 1 0 0 1 」とする。まず、ノード 2 1 3 は、自身の論理識別子 m が論理識別子 K e y と同じであるか否か判定する (ステップ S 1 0 1) 。ノード 2 1 3 の論理識別子の値は「 1 1 0 1 」であり、これは論理識別子 K e y の値「 1 0 0 1 」と異なることから (ステップ S 1 0 1 の N o) 、ノード 2 1 3 は、キーシフト k s h i f t の先頭ビット b と、論理識別子空間 N (ここでは、4 ビットであるため 1 6) とを用いて、 $2m + b \text{ mod } N$ で表わされる論理識別子を、ノード t とする (ステップ S 1 0 3) 。具体的には、キーシフト「 1 0 0 1 」の先頭ビット 1 と論理識別子空間 1 6 とを用いて、ノード t を $2 \times 13 + 1 \text{ mod } 16 = 11$ (2 進数表示で 1 0 1 1) とする。この演算は、上述したように、2 進数表示では m のビットを左ビットシフトさせ、最下位ビットに、キーの先頭ビットを与える処理と同等である。

【 0 0 1 3 】

次に、ノード 2 1 3 は、キーシフト k s h i f t の値「 1 0 0 1 」を左ビットシフトさせ、キーシフト k s h i f t の値を「 0 0 1 0 」とする (ステップ S 1 0 4) 。次に、論理識別子が 1 1 (2 進数表示で 1 0 1 1) であるノード 2 1 1 が、論理識別子 K e y の値「 1 0 0 1 」を、キーシフト k s h i f t の値「 0 0 1 0 」で探す処理に移行する (ステップ S 1 0 5) 。これは、図 1 7 に示す処理 (ステップ S 1 0 1 ~ S 1 0 5) がノード 2 1 1 に移されたことに相当し、ノード 2 1 3 にて実行したステップ S 1 0 5 は、そのサブルーチンとして、ノード 2 1 1 にてステップ S 1 0 1 ~ S 1 0 5 の処理が実行されることとなる。

【 0 0 1 4 】

ノード 2 1 1 では、データの論理識別子 K e y の値「 1 0 0 1 」とノード 2 1 1 の自身の論理識別子「 1 0 1 1 」が異なることから (ステップ S 1 0 1 の N o) 、キーシフト「 0 0 1 0 」の先頭ビット 0 と論理識別子空間 1 6 とからノード t の値を「 0 1 1 0 」とし (ステップ S 1 0 3) 、キーシフト k s h i f t の値を「 0 1 0 0 」とする (ステップ S 1 0 4) 。そして、ノード 2 1 1 は、論理識別子「 0 1 1 0 」を持つ新たなノード 2 0 6 に処理を移行させる (ステップ S 1 0 5) 。これにより、ノード 2 0 6 が、論理識別子 K e y の値「 1 0 0 1 」を、キーシフト k s h i f t の値「 0 1 0 0 」で探すこととなる。

【 0 0 1 5 】

ノード 2 0 6 では、データの論理識別子 K e y の値「 1 0 0 1 」とノード 2 0 6 の自身の論理識別子「 0 1 1 0 」が異なるため (ステップ S 1 0 1 の N o) 、キーシフト「 0 1 0 0 」の先頭ビット 0 と論理識別子空間 1 6 とからノード t の値を「 1 1 0 0 」とし (ステップ S 1 0 3) 、キーシフト k s h i f t の値を「 1 0 0 0 」とする (ステップ S 1 0 4) 。そして、ノード 2 0 6 は、論理識別子「 1 1 0 0 」を持つ新たなノード 2 1 2 に処理を移行させる (ステップ S 1 0 5) 。これにより、ノード 2 1 2 が、論理識別子 K e y の値「 1 0 0 1 」を、キーシフト k s h i f t の値「 1 0 0 0 」で探すこととなる。

【 0 0 1 6 】

ノード 2 1 2 では、データの論理識別子 K e y の値「 1 0 0 1 」とノード 2 1 2 の自身の論理識別子「 1 1 0 0 」が異なるため (ステップ S 1 0 1 の N o) 、キーシフト「 1 0 0 0 」の先頭ビット 1 と論理識別子空間 1 6 とからノード t の値を「 1 0 0 1 」とし (ステップ S 1 0 3) 、キーシフト k s h i f t の値を「 0 0 0 0 」とする (ステップ S 1 0

4)。そして、ノード212は、論理識別子「1001」を持つ新たなノード209に処理を移行させる(ステップS105)。これにより、ノード209が、論理識別子Keyの値「1001」を、キーシフトkshiftの値「0000」で探すこととなる。

【0017】

ノード209では、データの論理識別子Keyの値「1001」とノード209の自身の論理識別子「1001」が同一であるため(ステップS101のYes)、ノード209は、自身が当該論理識別子を持つデータを管理するノードであると判断する。そして、ノード209は、サブルーチンとして呼び出したノード212のステップS105の結果として論理識別子「1001」のノードであることを示すメッセージをノード212に返す(ステップS106)。

10

【0018】

ノード212のステップS106では、この結果をさらにサブルーチンとして呼び出したノード206に返し、ノード206のステップS106では、この結果をサブルーチンとして呼び出したノード211に返し、ノード211のステップS106では、この結果をサブルーチンとして呼び出したノード213に返す。このように、最終的に検索を実行したノード213に目的のノード209が返ることになる。

【0019】

このような手順を取ることで、図18のグラフに示すように、ノードが他のノードを指す数である次数(各ノードに隣接するノードの数、すなわち各ノードがリンクしているノードの数)が一定数(2つ)でありながら、全ノード数Nに対して、 $\log N$ の転送回数(ホップ数)で、特定のデータのKeyを管理するノードを発見することができる。

20

【0020】

(2)第2の従来技術について：

第1の従来技術はオーバレイネットワークに関係なく、de Bruijnグラフを用いた理論上のアルゴリズムについての説明であったが、実際には、Chordのハッシュ空間では各ノードは間隔を持って存在しているため(すなわち、論理識別子空間16(2の4乗)である場合に、論理識別子0~15の全てにノードが存在するわけではないため)、全ての論理識別子を持つノードが定義されたde Bruijnグラフは適用できない。そこで、Koordではde Bruijn上のIDをChordハッシュ空間上の仮想IDとし、一つのノードがある区間のIDを担当させている。このようにオーバレイネットワークにKoordを実装した場合の構成についても、非特許文献1に記載されている。

30

【0021】

図19は、第2の従来技術のシステム構成を示すブロック図である。また、図20は、第2の従来技術のデータ検索手順を示すフローチャートである。また、図21は、第2の従来技術におけるピア間でのメッセージの流れの具体例を示す説明図である。

【0022】

図19に示すように、第2の従来技術におけるデータ検索システムは、ネットワーク300上でのアドレスを有する複数のピア310、320、330から構成される。なお、ピア310、320、330は、それぞれ第1の従来技術におけるノードに相当する。これらのピア310、320、330はそれぞれ異なる論理識別子を有する。さらに、各ピア310、320、330は、ピア論理識別子格納部318と、ローカルデータ格納部311と、ルーティングテーブル312と、メッセージ転送手段313と、通信手段314と、左ビットシフト手段315と、先頭ビット取得手段316と、登録・検索実行手段317とを備えている。

40

【0023】

ピア論理識別子格納部318には、オーバレイネットワークにおいてピア310を他のピアと識別する論理識別子が格納される。ローカルデータ格納部311には、オーバレイネットワーク内のピアで共有されるデータのうち、ピア310が管理するデータが格納される。

50

【 0 0 2 4 】

ルーティングテーブル 3 1 2 には、他ピアの論理識別子と、そのピアにアクセスするために必要なネットワーク 3 0 0 上でのアドレスとが格納されている（図 2 1 に示すテーブル 5 0 0 参照）。このアドレスとしては、例えば Internet Protocol (IP) アドレスが用いられてもよい。

【 0 0 2 5 】

ルーティングテーブルには、d ノードと s ノードが格納される。d ノードは、各自のピアの論理識別子を m とした場合に、論理識別子 2 m の「predecessor」に対応するノード（ピア）である。ここで、論理識別子 2 m の「predecessor」とは、図 2 1 に示すリングの中で、論理識別子 2 m から見て反時計回りの方向に存在する最初のノード（ピア）である。この predecessor は、論理識別子 2 m との間のハッシュ空間を責任領域としてデータの管理を担当する。ただし、ここでの論理識別子 2 m は、厳密には論理識別子空間のサイズを N とした場合に、合同式 $m \bmod N$ と表される。図 2 1 に示す例では、ピア 5 1 6 の論理識別子が 1 3（2 進数表示で 1 1 0 1）であるため、d ノードは論理識別子 1 0（2 進数表示で 1 0 1 0）のピア（ノード）5 1 5 となる。

10

【 0 0 2 6 】

また、s ノードは、各自のピアの論理識別子を m とした場合に、論理識別子 m の「successor」に対応するノード（ピア）である。ここで、論理識別子 m の「successor」とは、図 2 1 に示すリングの中で、論理識別子 m から見て時計回りに存在する最初のノード（ピア）である。図 2 1 に示す例では、ピア 5 1 6 の論理識別子が 1 3（2 進数表示で 1 1 0 1）であるため、s ノードは論理識別子 1 5（2 進数表示で 1 1 1 1）のピア（ノード）5 1 7 となる。この例では、論理識別子サイズ N が 1 6 であるが、これを超える場合には、合同式 $m \bmod N$ を用いて、 $2 m \bmod N$ として処理される。

20

【 0 0 2 7 】

左ビットシフト手段 3 1 5 は、K o o r d e のアルゴリズムにもとづく左ビットシフトの処理（m のビット列を 1 ビットだけ左にシフトさせ、最下位ビットにキーの先頭ビットを与える処理）を実行する。先頭ビット取得手段 3 1 6 は、キーシフトの先頭ビットを取得し、取得した先頭ビットと論理識別子空間とから仮想ノード i を算出する処理を実行する。

30

【 0 0 2 8 】

登録・検索実行手段 3 1 7 は、データの登録および検索の処理を実行する。メッセージ転送手段 3 1 3 は、データの登録要求および検索メッセージ（検索要求）の転送処理を実行する。通信手段 3 1 4 は、ネットワーク 3 0 0 を通じて他のピアとの間で検索メッセージ等を送受信する処理を実行する。

【 0 0 2 9 】

このような構成を有する従来のデータ検索システムは次のように動作する。

【 0 0 3 0 】

あるピア m（例えば図 2 1 に示す論理識別子 1 1 0 1 のピア 5 1 6）の登録・検索実行手段 3 1 7 は、データの登録・検索要求が外部（例えば外部ユーザインタフェース）から与えられると、ハッシュ関数などを用いて、このデータが格納される 1 つの論理識別子を算出する。この論理識別子を論理識別子 key とキーシフト k s h i f t とし、当該ピアの論理識別子に 1 を加えた値を仮想ノード i とする。例えば“Foo”というデータの登録先を探す場合において、このハッシュ値が 0 1 1 1（2 進数表示）となったとする。これらの値を引数として、登録・検索実行手段 3 1 7 は、メッセージ転送手段 3 1 3 を呼び出す。

40

【 0 0 3 1 】

メッセージ転送手段 3 1 3 において、図 2 0 に示すフローチャートの各処理が実行される。上記の例では、まず、論理識別子 key の値「0 1 1 1」、キーシフト k s h i f t の値「0 1 1 1」、仮想ノード「1 1 1 0」がメッセージ転送手段 3 1 3 に与えられる。

50

【 0 0 3 2 】

メッセージ転送手段 3 1 3 は、論理識別子 key が m より大きく、 $successor$ 以下の範囲に含まれるか否かを判定する（ステップ S 4 0 1）。この範囲に含まれる場合は（ステップ S 4 0 1 の Yes）、 $successor$ が当該論理識別子 key を担当すると判別される。この場合は、検索要求に対する検索結果が検索元のピアに返される（ステップ S 4 0 2）。上記の範囲に含まれない場合は（ステップ S 4 0 1 の No）、メッセージ転送手段 3 1 3 は、仮想ノード i が上記の範囲（ m より大きく $successor$ 以下の範囲）に含まれるか否かを判定する（ステップ S 4 0 3）。

【 0 0 3 3 】

仮想ノード i が上記の範囲に含まれない場合は（ステップ S 4 0 3 の No）、メッセージ転送手段 3 1 3 は、当該ピア m にて処理が実行されたときの引数と同一の引数（論理識別子 key 、キーシフト $ks h i f t$ 、仮想ノード i ）で、当該ピア m の $successor$ を呼び出す（ステップ S 4 0 7）。これにより、 $successor$ が、論理識別子 Key を、キーシフト $ks h i f t$ および仮想ノード i で探す処理（ステップ S 4 0 1 ~ S 4 0 7）を実行する。

10

【 0 0 3 4 】

仮想ノード i が上記の範囲に含まれる場合は（ステップ S 4 0 3 の Yes）、ピア m の先頭ビット取得手段 3 1 6 は、キーシフト $ks h i f t$ の先頭ビット b と、当該仮想ノード i を用いて、 $2i + b \text{ mod } N$ の計算を行い、計算結果の論理識別子を仮想ノード i とする（ステップ S 4 0 4）。ここで、 N は論理識別子空間のサイズある。

20

【 0 0 3 5 】

次に、左ビットシフト手段 3 1 5 は、キーシフト $ks h i f t$ を 1 ビットだけ左にシフトさせる（ステップ S 4 0 5）。このようにして得られた論理識別子 key とキーシフト $ks h i f t$ 、仮想ノード i を引数として、ピア m の d ノードであるピアを呼び出す（ステップ S 4 0 6）。これにより、 d ノードであるピアが、論理識別子 Key を、キーシフト $ks h i f t$ および仮想ノード i で探す処理（ステップ S 4 0 1 ~ S 4 0 7）を実行する。

【 0 0 3 6 】

上記の例では、図 2 1 に示すように、最初の論理識別子が「1 1 0 1」であるピア 5 1 6 にて、論理識別子 key の値「0 1 1 1」、キーシフト $ks h i f t$ の値「0 1 1 1」、過疎ノード i 「1 1 1 0」でメッセージ転送手段 3 1 3 が呼び出される（図 2 1 の 5 2 1）。この場合、仮想ノード i が範囲（1 1 0 1, 1 1 1 1】に存在するので（ステップ S 4 0 3 の Yes）、仮想ノード i が「1 1 0 0」と算出され（ステップ S 4 0 4）、キーシフト $ks h i f t$ が「1 1 1 0」に変換されて（ステップ S 4 0 5）、 d ノードであるピア 5 1 5 にメッセージが転送される（ステップ S 4 0 6；図 2 1 の 5 2 2）。

30

【 0 0 3 7 】

ピア 5 1 5 においても、仮想ノード i 「1 1 0 0」が、範囲（1 0 1 0, 1 1 0 1】に存在するため（ステップ S 4 0 3 の Yes）、仮想ノード i が「1 0 0 1」と算出され（ステップ S 4 0 4）、キーシフト $ks h i f t$ が「1 1 0 0」に変換されて（ステップ S 4 0 5）、 d ノードであるピア 5 1 2 にメッセージが転送される（ステップ S 4 0 6；図 2 1 の 5 2 3）。

40

【 0 0 3 8 】

ピア 5 1 2 では、論理識別子 key の値「0 1 1 1」が範囲（1 0 1 0, 1 1 0 1】に存在せず（ステップ S 4 0 1 の No）、仮想ノード i 「1 0 0 1」も範囲（1 0 1 0, 1 1 0 1】に存在しないため（ステップ S 4 0 3 の No）、ピア 5 1 2 の $successor$ であるピア 5 1 3 にメッセージが転送される（ステップ S 4 0 7）。このピア 5 1 3 では、論理識別子 key の値「0 1 1 1」が範囲（0 1 1 0, 1 0 0 1】に存在するため（ステップ S 4 0 1 の Yes）、その $successor$ が論理識別子 key の値「0 1 1 1」を管理していると判定する。そして、ピア 5 1 3 のメッセージ転送手段 3 1 3 は、検索結果を検索元のピア 5 1 6 に返す（ステップ S 4 0 2）。

50

【0039】

このようにして、この第2の従来技術では、ノード数が図18に示したように論理識別子空間のサイズ分存在しない場合でも、定数次数と対数ホップ数を実現している。なお、定数次数とは全てのピアが他ピアのアドレスを認識する数は、ピアの総数によらず一定であることであり、対数ホップ数とは、あるピアからメッセージを転送して、あて先に到達するまでのホップ数が、ピアの総数 n としたときに $O(\log(n))$ であることを指す。

【0040】

【非特許文献1】M. Frans Kaashoek and David R. Karger, "Koorde: A simple degree-optimal distributed hash table"

10

【発明の開示】

【発明が解決しようとする課題】

【0041】

しかし、上述した従来のデータ検索方式では、完全一致検索しかできず、範囲検索ができない。従来のデータ検索方式は、データ格納先の論理識別子が一つのみであり、複数の論理識別子を含む集合を扱うことができないためである。

【0042】

本発明は、上記のような課題を解決するためになされたものであり、定数次数と対数ホップ数を実現するデータ検索方式でありながら範囲検索ができるデータ検索装置、データ検索方法およびデータ検索用プログラムを提供することを目的とする。

20

【課題を解決するための手段】

【0043】

以上の目的を達成するため、本発明のデータ検索装置は、ネットワーク上に配置され、他の装置と共有するデータを検索するデータ検索装置であって、他のデータ検索装置と区別するために付与される識別子であって、複数ビットで構成された論理識別子が付与され、検索対象となる各データ検索装置の論理識別子の集合である論理識別子範囲を指定した検索要求を受け取ると、論理識別子範囲に含まれる各論理識別子のうちの最大値である最大論理識別子と最小値である最小論理識別子とに基づいて、指定された論理識別子範囲を分割する分割処理を実行する分割手段と、分割手段によって分割された各論理識別子範囲を検索対象として指定した検索要求を、転送先の装置に転送する検索要求転送手段と、該データ検索装置と、該データ検索装置の論理識別子に対して時計回りの次の論理識別子に対応する装置である Successor との間に位置するノードから任意に選択した仮想ノードの識別子である仮想識別子を生成する検索実行手段と、分割手段は、最大論理識別子と最小論理識別子に含まれる先頭ビットが同一である場合に、指定された論理識別子範囲に含まれる各論理識別子を先頭ビットを押し出すようにビットシフトしたものを1つの論理識別子範囲とする処理を実行し、最大論理識別子と最小論理識別子に含まれる先頭ビットが異なる場合に、指定された論理識別子範囲における各論理識別子の先頭ビットを押し出すようにビットシフトしたものの集合についてシフト後の最小論理識別子以上の各論理識別子を含む論理識別子範囲とシフト後の最大論理識別子以下の各論理識別子を含む論理識別子範囲とを得るビットシフト範囲分割処理を実行し、分割手段は、指定された論理識別子範囲を Successor の論理識別子を境界にして2つに分割する範囲分割処理を実行する範囲分割手段を含み、該データ検索装置の論理識別子が表す数値と Successor の論理識別子が表す数値とで示される範囲である特定範囲に仮想ノードの論理識別子が含まれる場合に、範囲分割手段によって分割された各論理識別子範囲のうち論理識別子が表す数値が大きい方の論理識別子範囲に対してビットシフト範囲分割処理を実行し、検索要求転送手段は、範囲分割手段によって分割された各論理識別子範囲を検索対象として指定するとともに仮想ノードの論理識別子を含む検索要求を、論理識別子範囲に含まれる装置に転送することを特徴とする。

30

40

【0047】

50

分割手段は、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子が特定範囲に含まれる場合に、範囲分割処理およびビットシフト範囲分割処理を実行するように構成されていてもよい。

【0048】

分割手段は、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子が特定範囲に含まれない場合、検索要求を検索要求転送手段に送るように構成されていてもよい。

【0049】

検索実行手段は、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子が特定範囲に含まれない場合、検索要求を検索要求転送手段に送るように構成されていてもよい。

10

【0050】

データ検索装置は、分割手段によって分割された論理識別子範囲毎に、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子と該論理識別子範囲に含まれる各論理識別子の先頭のビットとから仮想ノードの論理識別子を算出する先頭ビット取得手段を備えていてもよい。

【0051】

分割手段は、範囲分割処理によって分割された論理識別子範囲のうち特定範囲に含まれない論理識別子範囲を、あらかじめ設定された仮想範囲をもとに分割する第2分割手段と、第2分割手段によって分割された論理識別子範囲のうち特定範囲に含まれる論理識別子範囲を左ビットシフトするとともに、シフト回数を増加させる範囲左ビットシフト手段とを含み、検索要求転送手段は、検索対象の論理識別子範囲とともに仮想範囲とシフト回数を含む検索要求を転送するように構成されていてもよい。

20

【0053】

本発明のデータ検索方法は、ネットワーク上に配置され、他の装置と共有するデータを検索するデータ検索装置で実行されるデータ検索方法であって、データ検索装置には、他のデータ検索装置と区別するために付与される識別子であって、複数ビットで構成された論理識別子が付与され、検索対象となる各データ検索装置の論理識別子の集合である論理識別子範囲を指定した検索要求を受け取ると、論理識別子範囲に含まれる各論理識別子のうちの最大値である最大論理識別子と最小値である最小論理識別子とに基づいて、指定された論理識別子範囲を分割する分割ステップと、分割ステップによって分割された各論理識別子範囲を検索対象として指定した検索要求を、転送先の装置に転送する検索要求転送ステップと、該データ検索装置と、該データ検索装置の論理識別子に対して時計回りの次の論理識別子に対応する装置であるSuccessorとの間に位置するノードから任意に選択した仮想ノードの識別子である仮想識別子を生成する検索実行ステップとを実行し、分割ステップで、最大論理識別子と最小論理識別子に含まれる先頭ビットが同一である場合に、指定された論理識別子範囲に含まれる各論理識別子を先頭ビットを押し出すようにビットシフトしたものを1つの論理識別子範囲とする処理を実行し、最大論理識別子と最小論理識別子に含まれる先頭ビットが異なる場合に、指定された論理識別子範囲における各論理識別子の先頭ビットを押し出すようにビットシフトしたものの集合についてシフト後の最小論理識別子以上の各論理識別子を含む論理識別子範囲とシフト後の最大論理識別子以下の各論理識別子を含む論理識別子範囲とを得るビットシフト範囲分割ステップを実行し、分割ステップは、指定された論理識別子範囲をSuccessorの論理識別子を境界にして2つに分割する範囲分割ステップを含み、分割ステップで、該データ検索装置の論理識別子が表す数値とSuccessorの論理識別子が表す数値とで示される範囲である特定範囲に仮想ノードの論理識別子が含まれる場合に、範囲分割ステップによって分割された各論理識別子範囲のうち論理識別子が表す数値が大きい方の論理識別子範囲に対してビットシフト範囲分割ステップを実行し、検索要求転送ステップで、範囲分割ステップによって分割された各論理識別子範囲を検索対象として指定するとともに仮想ノードの論理識別子を含む検索要求を、論理識別子範囲に含まれる装置に転送することを特

30

40

50

徴とする。

【 0 0 5 5 】

分割ステップで、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子が特定範囲に含まれる場合に、範囲分割ステップおよびビットシフト範囲分割ステップを実行するようにしてもよい。

分割ステップで、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子が特定範囲に含まれない場合、検索要求を検索要求転送ステップに送るようにしてもよい。

検索実行ステップで、仮想ノードの論理識別子に含まれる後尾のビットが、受け取った検索要求で指定される論理識別子範囲に含まれる各論理識別子の特定の位置のビットとなるように仮想ノードの論理識別子を生成するようにしてもよい。

10

データ検索装置は、分割ステップによって分割された論理識別子範囲毎に、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子と該論理識別子範囲に含まれる各論理識別子の先頭のビットとから仮想ノードの論理識別子を算出するようにしてもよい。

分割ステップは、範囲分割ステップによって分割された論理識別子範囲のうち特定範囲に含まれない論理識別子範囲を、あらかじめ設定された仮想範囲をもとに分割する第2分割ステップと、第2分割ステップによって分割された論理識別子範囲のうち特定範囲に含まれる論理識別子範囲を左ビットシフトするとともに、シフト回数を増加させる範囲左ビットシフトステップとを含み、検索要求転送ステップで、検索対象の論理識別子範囲とともに仮想範囲とシフト回数を含む検索要求を転送するようにしてもよい。

20

【 0 0 5 6 】

本発明のデータ検索プログラムは、検索対象となる各データ検索装置の論理識別子の集合である論理識別子範囲を指定した検索要求を受け取ると、論理識別子範囲に含まれる各論理識別子のうちの最大値である最大論理識別子と最小値である最小論理識別子とに基づいて、指定された論理識別子範囲を分割する分割処理と、分割処理によって分割された各論理識別子範囲を検索対象として指定した検索要求を、転送先の装置に転送する検索要求転送処理と、該データ検索装置と、該データ検索装置の論理識別子に対して時計回りの次の論理識別子に対応する装置である S u c c e s s o r との間に位置するノードから任意に選択した仮想ノードの識別子である仮想識別子を生成する検索実行処理とを実行させ、分割処理で、最大論理識別子と最小論理識別子に含まれる先頭ビットが同一である場合に、指定された論理識別子範囲に含まれる各論理識別子を先頭ビットを押し出すようにビットシフトしたものを1つの論理識別子範囲とする処理を実行し、最大論理識別子と最小論理識別子に含まれる先頭ビットが異なる場合に、指定された論理識別子範囲における各論理識別子の先頭ビットを押し出すようにビットシフトしたものの集合についてシフト後の最小論理識別子以上の各論理識別子を含む論理識別子範囲とシフト後の最大論理識別子以下の各論理識別子を含む論理識別子範囲とを得るビットシフト範囲分割処理を実行させ、分割処理は、指定された論理識別子範囲を S u c c e s s o r の論理識別子を境界にして2つに分割する範囲分割処理を含み、分割処理で、該データ検索装置の論理識別子が表す数値と S u c c e s s o r の論理識別子が表す数値とで示される範囲である特定範囲に仮想ノードの論理識別子が含まれる場合に、範囲分割処理によって分割された各論理識別子範囲のうち論理識別子が表す数値が大きい方の論理識別子範囲に対してビットシフト範囲分割処理を実行させ、検索要求転送処理で、範囲分割処理によって分割された各論理識別子範囲を検索対象として指定するとともに仮想ノードの論理識別子を含む検索要求を、論理識別子範囲に含まれる装置に転送させることを特徴とする。

30

40

【 0 0 5 8 】

分割処理で、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子が特定範囲に含まれる場合に、範囲分割処理およびビットシフト範囲分割処理を実行させるようにしてもよい。

分割処理で、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの

50

論理識別子が特定範囲に含まれない場合、検索要求を検索要求転送処理に送らせるようにしてもよい。

検索実行処理で、仮想ノードの論理識別子に含まれる後尾のビットが、受け取った検索要求で指定される論理識別子範囲に含まれる各論理識別子の特定の位置のビットとなるように仮想ノードの論理識別子を生成させるようにしてもよい。

分割処理によって分割された論理識別子範囲毎に、他のデータ検索装置から転送されてきた検索要求に含まれる仮想ノードの論理識別子と該論理識別子範囲に含まれる各論理識別子の先頭のビットとから仮想ノードの論理識別子を算出させるようにしてもよい。

分割処理で、範囲分割処理によって分割された論理識別子範囲のうち特定範囲に含まれない論理識別子範囲を、あらかじめ設定された仮想範囲をもとに分割する第2分割処理と、第2分割処理によって分割された論理識別子範囲のうち特定範囲に含まれる論理識別子範囲を左ビットシフトするとともに、シフト回数を増加させる範囲左ビットシフト処理とを実行させ、検索要求転送処理で、検索対象の論理識別子範囲とともに仮想範囲とシフト回数を含む検索要求を転送させるようにしてもよい。

10

【発明の効果】

【0059】

以上のように、本発明によれば、定数次数と対数ホップ数を有するオーバレイデータ検索方式において、データの範囲検索を確実に行うことができる。範囲検索を実現するには、検索実行者が指定した検索式を満たす要素の集合（論理識別子範囲）を検索対象とする必要があるが、複数の論理識別子を含んだ範囲を適宜分割しながら、当該範囲に属する論理識別子を担うピア（ノード）に到達するように、メッセージ（検索要求）を転送できるためである。

20

【発明を実施するための最良の形態】

【0060】

以下、本発明の実施の一形態を図面を参照して説明する。

【0061】

実施の形態1.

図1は、この実施の形態1におけるデータ検索システムの構成を示すブロック図である。図1に示すように、実施の形態1におけるデータ検索システムは、ネットワーク100上でのアドレスを有する複数のノード110、120、130から構成される。図1には、ネットワーク100に接続されているノードはノード110、120、130だけであるが、ほかにも複数のノードがネットワーク100に接続されている。各ノードは、例えばサーバやパーソナルコンピュータなどの計算機（端末）で実現されている。

30

【0062】

図1に示すように、ノード110は、通信インタフェース（通信IF）111と、ユーザインタフェース（ユーザIF）112と、ハードディスク（HD）113と、メモリ114と、CPU115とを備えている。なお、ノード110以外のノード120、130も、ノード110と同一の構成を備えている。また、通信IF111やユーザIF112などの各部の構成はバスで接続されている。

【0063】

図1に示す構成において、通信IF111は、ネットワーク100と接続され、データの送受信を制御するインタフェースである。ユーザIF112は、ユーザに対する情報の表示や、ユーザのデータ入力を受け付けるインタフェースである。図1に示すように、ユーザIF112は、ユーザが入力操作可能な入力部116と接続され、入力部116からのデータ入力を受け付ける。なお、図1に示す例では、ノード110と入力部116とを別構成としているが、入力部116がノード110の構成として設けられていてもよい。

40

【0064】

HD113は、制御プログラム（データ検索用プログラム）や各種データを格納する記憶媒体（外部記憶装置）である。なお、図1に示す例では、記憶媒体としてHD113を示しているが、光磁気ディスクなどの記憶媒体であってもよい。メモリ114は、CPU

50

115が直接アクセスする制御プログラムやデータを記憶する記憶装置（主記憶装置）である。CPU115は、制御プログラムに従って各々の処理を実行する中央処理装置である。

【0065】

次に、この実施の形態1におけるデータ検索システムの動作について説明する。

【0066】

なお、この実施の形態1において、論理識別子範囲は、範囲検索対象の論理識別子の範囲である。そして、最大論理識別子とは、当該論理識別子範囲の上限値に相当する。また、最小論理識別子とは、当該論理識別子範囲の下限値に相当する。また、シフト後論理識別子範囲は、左ビットシフトされた後の論理識別子範囲である。そして、シフト後最大論理識別子は、最大論理識別子をビットシフトして得られる値である。また、シフト後最小論理識別子は、最小論理識別子をビットシフトして得られる値である。

10

【0067】

図2は、この実施の形態1のデータ検索手順を示すフローチャートである。なお、図2に示すデータ検索手順は、ノードmにおいて、論理識別子範囲idRangesを探す手順が非同期処理であるとした場合の処理手順である。

【0068】

ユーザは、論理識別子範囲idRangesで指定される範囲のデータを取得する場合、入力部116を介して論理識別子範囲idRangesを指定した検索要求をノードm（論理識別子mのノード）に入力する。そして、検索要求に応じて、ノードmにおいて図2に示すデータ検索処理が実行される。

20

【0069】

図2に示すデータ検索手順が実行されると、まず、ノードmのCPU115は、論理識別子範囲idRangesに含まれる論理識別子が当該ノードmの論理識別子のみが含まれるか否かを判定する（ステップS701）。この判定は、論理識別子範囲に含まれる最大論理識別子と最小論理識別子を用いて判定される。判定結果として、ノードmの論理識別子のみであると判定された場合は（ステップS701のYes）、ノードmのCPU115は、自身が検索要求された論理識別子範囲のデータを持つノードであると判断する。そして、ノードmのCPU115は、検索要求に応じて図2に示す処理を初期に実行したノードに、ノードmを検索結果として返す（ステップS702）。

30

【0070】

なお、ここでは、図2に示す処理を初期に実行したノードが自身であるため、図2に示す処理を初期に実行したノードへのポインタ情報を省略しているが、図2に示すデータ検索処理の実行に当たっては、そのポインタ情報が引き継がれるものとし、ステップS702ではそのポインタ情報を用いて、初期のノードに対して検索結果が返される。

【0071】

ステップS701の判定結果にて、mの論理識別子のみではないと判定された場合には（ステップS701のNo）、ノードmのCPU115は、論理識別子範囲idRangesに対して、図3のフローチャートで示される左ビットシフト範囲分割処理の手順を実行し、分割後の論理識別子範囲Range0と論理識別子範囲Range1を取得する（ステップS703）。

40

【0072】

ステップS703で取得された論理識別子範囲Range0がNullでない場合には（ステップS704のNo）、ノードmのCPU115は、ノードmの0側次ノードを取り出し、このノードにて、論理識別子範囲Range0を探す処理を非同期に実行する（ステップS705）。このステップS705の処理は、図2に示すデータ検索処理の手順と同一であり、再帰的に実行される。

【0073】

ステップS705の処理を非同期実行後、あるいはステップS704にて論理識別子範囲Range0がNullであるとされた場合には（ステップS704のYes）、ノード

50

ノードmのCPU115は、論理識別子範囲Range1がNullであるか否かを判定する(ステップS706)。ここで、論理識別子範囲Range1がNullであると判定しなかった場合には(ステップS706のNo)、ノードmのCPU115は、ノードmの1側次ノードを取り出し、このノードにて、論理識別子範囲Range1を探す処理を非同期に実行する(ステップS707)。このステップS707の処理も、図2に示すデータ検索処理の手順と同一であり、再帰的に実行される。

【0074】

ステップS707の処理を非同期実行後、あるいはステップS706にて論理識別子範囲Range1がNullであると判定された場合には(ステップS706のYes)、図2に示す処理が終了する。このような手順により、論理識別子範囲idRangesは逐次分割され、目的とするノードに到達すると、その結果が初期に実行したノードに返される。

10

【0075】

図3は、左ビットシフト範囲分割処理の手順を示すフローチャートである。図3に示す左ビットシフト範囲分割処理において、ノードmのCPU115は、論理識別子範囲idRangesが与えられると、論理識別子範囲idRangesに含まれる、ビットシフト対象の最大論理識別子とビットシフト対象の最小論理識別子とを取り出し(ステップS601)、これらの先頭ビットが同一か否かを判定する(ステップS602)。

【0076】

同一である場合に(ステップS602のYes)、その先頭ビットが1である場合には(ステップS606のYes)、ステップS607の処理を実行して、処理を終了する。先頭ビットが0である場合には(ステップS606のNo)、ステップS608の処理を実行して、処理を終了する。

20

【0077】

先頭ビットが1である場合のステップS607の処理では、CPU115は、論理識別子範囲idRanges内の論理識別子を全て左ビットシフトして、新たな論理識別子範囲Range1に与え、他の新たな論理識別子範囲Range0にはNullに与える。

【0078】

先頭ビットが0である場合のステップS608の処理では、CPU115は、論理識別子範囲idRanges内の論理識別子を全て左ビットシフトして、新たな論理識別子範囲Range0に与え、他の新たな論理識別子範囲Range1にはNullを与える。

30

【0079】

なお、Nullが与えられた論理識別子範囲は実質存在しないことと同一であるため、ステップS602において先頭ビットが同一であると判定された場合は(ステップS602のYes)、与えられた論理識別子範囲idRangesは分割されないと解釈してよい。

【0080】

ステップS602において先頭ビットが同一でないと判定された場合には(ステップS602のNo)、CPU115は、論理識別子範囲idRangesに含まれるビットシフト対象の論理識別子範囲の先頭ビットが同一であるような複数集合(複数の論理識別子範囲)に分割し、これを返す。

40

【0081】

すなわち、図3に示すように、CPU115は、1側最小論理識別子と0側最大論理識別子を生成する(ステップS603)。ここで、1側最小論理識別子とは、先頭ビットが1でそれ以降のビットが0である論理識別子である。また、0側最大論理識別子とは、先頭ビットが0でそれ以降のビットが1である論理識別子である。

【0082】

次に、CPU115は、新たな論理識別子範囲Range1に1側最小論理識別子以上の論理識別子範囲idRangesを入力し、別の新たな論理識別子範囲Range0に0側最大論理識別子以下の論理識別子範囲idRangesを入力する(ステップS60

50

4)。このようにして、左ビットシフト範囲分割処理では、最大論理識別子と最小論理識別子の先頭ビットの違いにより論理識別子範囲を分割する。

【0083】

そして、CPU115は、新たな論理識別子範囲Range0,1の論理識別子をそれぞれ左ビットシフトする(ステップS605)。なお、図3に示す例では、論理識別子範囲idRangesを論理識別子範囲Range0,1に分割した後に左ビットシフトする処理を実行しているが、論理識別子範囲idRangesを左ビットシフトした後に論理識別子範囲Range0,1に分割するようにしてもよい。また、論理識別子範囲idRangesを論理識別子範囲Range0,1に分割すると同時に左ビットシフトする処理を実行するようにしてもよい。

10

【0084】

次に、この実施の形態1におけるデータ検索システムの動作の具体例(データ検索処理の手順の具体例)について、図4および図5を用いて説明する。

【0085】

図4は、この実施の形態1におけるde Bruijnグラフを示す説明図である。また、図5は、この実施の形態1における左ビットシフト範囲分割処理のアルゴリズムを視覚的に示す説明図である。

【0086】

図4に示すように、論理識別子が与えられたノードがネットワーク内に存在するものとし、論理識別子13(2進数表示で「1101」)であるノード413が、データの論理識別子範囲が[7,11]であるデータを取得するものとする。

20

【0087】

この論理識別子に相当するノードは、図4では、論理識別子7(2進数表示で「0111」)がノード407、論理識別子8(2進数表示で「1000」)がノード408、論理識別子9(2進数表示で「1001」)がノード409、論理識別子10(2進数表示で「1010」)がノード410、論理識別子11(2進数表示で「1011」)がノード411となる。これらのノードは、ネットワーク内に散在しているが、この実施の形態1で示した手順により、これら散在するノードからのデータ取得を可能とする。

【0088】

図5の1201に記載の論理識別子範囲は、検索要求された論理識別子範囲[7,11]であって、左ビットシフト前の最大論理識別子と左ビットシフト前の最小論理識別子のビット列を木構造で表記している。また、図5の1202~1213に記載の論理識別子範囲は、シフト後最大論理識別子とシフト後最小論理識別子のビット列を木構造で表記している。また、図5の1201~1213に記載の論理識別子範囲におけるMin.以降には、ビット列に対応する最小論理識別子の値の10進数表記と、その横の()内にシフトが行われていない最小論理識別子の値の10進数表記とを表記している。さらに、図5の1201~1213に記載の論理識別子範囲におけるMax.以降には、ビット列に対応する最大論理識別子の値の10進数表記と、その横の()内にシフトが行われていない最大論理識別子の値の10進数表記とを表記している。また、木構造のRootの左には、その木構造のビットシフト前の先頭ビットを示している。

30

40

【0089】

まず、ノード413では、図5の1201に記載の論理識別子範囲[7,11]を探す処理を行う。図2に示すステップS701において、ノード413のCPU115は、論理識別子範囲[7,11]内にノード413の論理識別子「1101」のみが含まれるかを判定する(ステップS701)。しかし、上記の例では、ノード413の論理識別子「1101」は論理識別子範囲[7,11](最小論理識別子「0111」と最大論理識別子「1011」の範囲)に含まれないので、ステップS701の条件は成り立たない(ステップS701のNo)。従って、ノード413のCPU115は、論理識別子範囲[7,11]に対して、左ビットシフトを行い、論理識別子範囲[7,7](図5の1203)と論理識別子範囲[8,11](図5の1202)に分割する(ステップS703

50

）。

【 0 0 9 0 】

具体的には、ノード 4 1 3 の CPU 1 1 5 が図 3 に示した処理を実行して論理識別子範囲を複数の論理識別子範囲に分割する。すなわち、ノード 4 1 3 の CPU 1 1 5 は、論理識別子範囲 [7 , 1 1] に含まれる最大論理識別子の先頭ビットと最小論理識別子の先頭ビットとを取り出す（ステップ S 6 0 1）。ここでは、最大論理識別子「 1 0 1 1 」の先頭ビット「 1 」が取り出され、最小論理識別子「 0 1 1 1 」の先頭ビット「 0 」が取り出される。

【 0 0 9 1 】

そして、CPU 1 1 5 は、最大論理識別子の先頭ビットと最小論理識別子の先頭ビットが同一であるか否かを判定する（ステップ S 6 0 2）。上記の例では、各々の先頭ビットが「 1 」と「 0 」であって同一でないので（ステップ S 6 0 2 の No）、CPU 1 1 5 は、1 側最小論理識別子と 0 側最大論理識別子を生成する（ステップ S 6 0 3）。ここで、1 側最小論理識別子は「 1 0 0 0 」であり、0 側最大論理識別子は「 0 1 1 1 」である。

【 0 0 9 2 】

次いで、CPU 1 1 5 は、新たな論理識別子範囲 Range 1 に、論理識別子範囲 [7 , 1 1] の中で 1 側最小論理識別子「 1 0 0 0 」以上の論理識別子範囲を入力し、別の新たな論理識別子範囲 Range 0 に、論理識別子範囲 [7 , 1 1] の中で 0 側最大論理識別子「 0 1 1 1 」以下の論理識別子範囲を入力する（ステップ S 6 0 4）。上記の例では、論理識別子範囲 [7 , 1 1] のうち、1 側最小論理識別子「 1 0 0 0 」以上の論理識別子範囲 [8 , 1 1] が新たな論理識別子範囲 Range 1 に入力される。また、論理識別子範囲 [7 , 1 1] のうち、0 側最大論理識別子「 0 1 1 1 」以下の論理識別子範囲 [7 , 7] が別の新たな論理識別子範囲 Range 0 に入力される。このようにして、左ビットシフト範囲分割処理では、先頭ビットの違いにより論理識別子範囲 [7 , 1 1] を論理識別子範囲 [7 , 7] と論理識別子範囲 [8 , 1 1] に分割する。

【 0 0 9 3 】

そして、ノード 4 1 3 の CPU 1 1 5 は、新たな論理識別子範囲 Range 0（論理識別子範囲 [7 , 7]）内の論理識別子を全て左ビットシフトするとともに、新たな論理識別子範囲 Range 1（論理識別子範囲 [8 , 1 1]）内の論理識別子を全て左ビットシフトする（ステップ S 6 0 5）。このようにして、左ビットシフトされた分割後の論理識別子範囲 Range 0（図 5 の 1 2 0 3）と論理識別子範囲 Range 1（図 5 の 1 2 0 2）を取得する。

【 0 0 9 4 】

次いで、ノード 4 1 3 の CPU 1 1 5 は、ステップ S 7 0 3 で分割された新たな論理識別子範囲 Range 0（論理識別子範囲 [7 , 7]）が Null であるかどうかを判定する（ステップ S 7 0 4）。上記の例では、論理識別子範囲 Range 0 が Null ではないので（ステップ S 7 0 4 の No）、CPU 1 1 5 は、ノード 4 1 3 の 0 側次ノードであるノード 4 1 0 にメッセージを転送して、そのノード 4 1 0 において論理識別子範囲 Range 0 を検索させる（ステップ S 7 0 5）。

【 0 0 9 5 】

また、ノード 4 1 3 の CPU 1 1 5 は、ステップ S 7 0 3 で分割された新たな論理識別子範囲 Range 1（論理識別子範囲 [8 , 1 1]）が Null であるかどうかを判定する（ステップ S 7 0 6）。この場合も、上記の例では、論理識別子範囲 Range 1 が Null ではないので（ステップ S 7 0 6 の No）、CPU 1 1 5 は、ノード 4 1 3 の 1 側次ノードであるノード 4 1 1 にメッセージを転送して、そのノード 4 1 1 において論理識別子範囲 Range 1 を検索させる（ステップ S 7 0 7）。

【 0 0 9 6 】

次に、ノード 4 1 1 では、図 5 の 1 2 0 2 に記載の論理識別子範囲 [8 , 1 1] を探す処理を行う。ノード 4 1 3 から論理識別子範囲 [8 , 1 1] の範囲検索要求を受け付けたノード 4 1 1 は、論理識別子範囲 [8 , 1 1] に自身のノードの論理識別子「 1 0 1 1 」

10

20

30

40

50

のみが含まれるかどうかを判定する（ステップS701）。

【0097】

論理識別子範囲[8, 11]はノード411の論理識別子「1011」のみを含むものではないため（ステップS701のNo）、ノード411のCPU115は、論理識別子範囲[8, 11]に対して、左ビットシフトを行い、複数の論理識別子範囲に分割する処理（図3に示す左ビットシフト範囲分割処理）を実行する（ステップS703）。しかし、ここでは、論理識別子範囲も上限値と下限値の先頭ビットが同一であるため、2つには分割されない。

【0098】

具体的には、ノード411のCPU115は、論理識別子範囲[8, 11]に含まれる最大論理識別子の先頭ビットと最小論理識別子の先頭ビットとを取り出す（ステップS601）。ここでは、図5の論理識別子範囲1202に示すように、最大論理識別子「0110」の先頭ビット「0」が取り出され、最小論理識別子「0000」の先頭ビット「0」が取り出される。

【0099】

そして、CPU115は、最大論理識別子の先頭ビットと最小論理識別子の先頭ビットが同一であるか否かを判定する（ステップS602）。上記の例では、各々の先頭ビットがいずれも「0」であって同一である（ステップS602のYes）。従って、CPU115は、先頭ビットが「0」であると判定し（ステップS606のNo）、論理識別子範囲（図5の1202）内の論理識別子を左ビットシフトする処理を実行し、左ビットシフトした論理識別子範囲を新たな論理識別子範囲Range0に入力し、別の新たな論理識別子範囲Range1にNullを入力する（ステップS608）。

【0100】

その後、ノード411のCPU115は、ビットシフトのみ行われた新たな論理識別子範囲Range0（図5の1204）がNullであるかどうかを判定する（ステップS704）。このとき、論理識別子範囲Range0はNullではないので（ステップS704のNo）、CPU115は、ノード411の0側次ノードであるノード406にメッセージを転送して、そのノード406において論理識別子範囲Range0を検索させる（ステップS705）。

【0101】

なお、ノード411のCPU115は、新たな論理識別子範囲Range1に対してもNullであるかどうかを判定するが（ステップS706）、論理識別子範囲Range1はNullであるため（ステップS706のYes）、ステップS707の処理を実行しない。

【0102】

次に、ノード410では、図5の1203に記載の論理識別子範囲[7, 7]を探す処理を行う。ノード413から論理識別子範囲[7, 7]の範囲検索要求を受け付けたノード410は、論理識別子範囲[7, 7]に自身のノードの論理識別子「1010」のみが含まれるかどうかを判定する（ステップS701）。

【0103】

論理識別子範囲[7, 7]はノード410の論理識別子「1010」のみを含むものではないため（ステップS701のNo）、ノード410のCPU115は、論理識別子範囲[7, 7]に対して、左ビットシフトを行い、複数の論理識別子範囲に分割する処理（図3に示す左ビットシフト範囲分割処理）を実行する（ステップS703）。しかし、ここでは、論理識別子範囲も上限値と下限値の先頭ビットが同一であるため、2つには分割されない。

【0104】

具体的には、ノード410のCPU115は、論理識別子範囲[7, 7]に含まれる最大論理識別子の先頭ビットと最小論理識別子の先頭ビットとを取り出す（ステップS601）。ここでは、図5の論理識別子範囲1203に示すように、最大論理識別子と最小論

10

20

30

40

50

理識別子はいずれも「1110」であり、その先頭ビット「1」が取り出される。

【0105】

そして、CPU115は、最大論理識別子の先頭ビットと最小論理識別子の先頭ビットが同一であるか否かを判定する(ステップS602)。上記の例では、各々の先頭ビットがいずれも「1」であって同一である(ステップS602のYes)。従って、CPU115は、先頭ビットが「1」であると判定し(ステップS606のYes)、論理識別子範囲(図5の1203)内の論理識別子を左ビットシフトする処理を実行し、左ビットシフトした論理識別子範囲を新たな論理識別子範囲Range1に入力し、別の新たな論理識別子範囲Range0にNullを入力する(ステップS607)。

【0106】

その後、ノード410のCPU115は、ビットシフトのみ行われた新たな論理識別子範囲Range1(図5の1205)がNullであるかどうかを判定する(ステップS706)。このとき、論理識別子範囲Range1はNullではないので(ステップS706のNo)、CPU115は、ノード410の1側次ノードであるノード405にメッセージを転送して、そのノード405において論理識別子範囲Range1を検索させる(ステップS707)。

【0107】

なお、ノード410のCPU115は、新たな論理識別子範囲Range0に対してもNullであるかどうかを判定するが(ステップS704)、論理識別子範囲Range0はNullであるため(ステップS704のYes)、ステップS705の処理を実行しない。

【0108】

次に、ノード406では、図5の1204に記載の論理識別子範囲[8, 11]を探す処理を行う。図2に示すステップS701において、ノード406のCPU115は、論理識別子範囲[8, 11]内にノード406の論理識別子「0110」のみが含まれるか否かを判定する(ステップS701)。しかし、上記の例では、論理識別子範囲[8, 11]はノード406の論理識別子「0110」のみを含むものではないので(ステップS701のNo)、ノード406のCPU115は、論理識別子範囲[8, 11]に対して、左ビットシフトを行い、論理識別子範囲[8, 9](図5の1207)と論理識別子範囲[10, 11](図5の1206)に分割する(ステップS703)。

【0109】

具体的には、ノード406のCPU115が図3に示した処理を実行して論理識別子範囲を複数の論理識別子範囲に分割する。すなわち、ノード406のCPU115は、論理識別子範囲[8, 11]に含まれる最大論理識別子の先頭ビットと最小論理識別子の先頭ビットとを取り出す(ステップS601)。ここでは、最大論理識別子「1000」の先頭ビット「1」が取り出され、最小論理識別子「0000」の先頭ビット「0」が取り出される。

【0110】

そして、CPU115は、最大論理識別子の先頭ビットと最小論理識別子の先頭ビットが同一であるか否かを判定する(ステップS602)。上記の例では、各々の先頭ビットが「1」と「0」であって同一でない(ステップS602のNo)、CPU115は、1側最小論理識別子と0側最大論理識別子を生成する(ステップS603)。

【0111】

ここで、1側最小論理識別子は「1010」(10進数表記で「10」となる。すなわち、図5の1204に記載の論理識別子範囲は以前に2回ビットシフトされている。2回のビットシフトで取り出された(はじき出された)値はそれぞれ「1」と「0」である(図5の1202で「1」が取り出され、図5の1204で「0」が取り出されている)。このことから、ビットシフトされる前の1側の論理識別子範囲の値は「101*」(「*」は「0」または「1」となる。従って、1側最小論理識別子は「1010」となる。

。

10

20

30

40

50

【 0 1 1 2 】

また、0側最大論理識別子は「1001」（10進数表記で「9」）となる。すなわち、図5の1204に記載の論理識別子範囲は以前に2回ビットシフトされている。2回のビットシフトで取り出された（はじき出された）値はそれぞれ「1」と「0」である（図5の1202で「1」が取り出され、図5の1204で「0」が取り出されている）。このことから、ビットシフトされる前の0側の論理識別子範囲の値は「100*」（「*」は「0」または「1」）となる。従って、0側最大論理識別子は「1001」となる。

【 0 1 1 3 】

次いで、CPU115は、新たな論理識別子範囲Range1に、論理識別子範囲[8, 11]の中で1側最小論理識別子「1010」以上の論理識別子範囲を入力し、別の新たな論理識別子範囲Range0に、論理識別子範囲[8, 11]の中で0側最大論理識別子「1001」以下の論理識別子範囲を入力する（ステップS604）。上記の例では、論理識別子範囲[8, 11]のうち、1側最小論理識別子「1010」以上の論理識別子範囲[10, 11]が新たな論理識別子範囲Range1に入力される。また、論理識別子範囲[8, 11]のうち、0側最大論理識別子「1001」以下の論理識別子範囲[8, 9]が別の新たな論理識別子範囲Range0に入力される。このようにして、左ビットシフト範囲分割処理では、先頭ビットの違いにより論理識別子範囲[8, 11]を論理識別子範囲[8, 9]と論理識別子範囲[10, 11]に分割する。

【 0 1 1 4 】

そして、ノード406のCPU115は、新たな論理識別子範囲Range0（論理識別子範囲[8, 9]）内の論理識別子を全て左ビットシフトするとともに、新たな論理識別子範囲Range1（論理識別子範囲[10, 11]）内の論理識別子を全て左ビットシフトする（ステップS605）。このようにして、左ビットシフトされた分割後の論理識別子範囲Range0（図5の1207）と論理識別子範囲Range1（図5の1206）を取得する。

【 0 1 1 5 】

次いで、ノード406のCPU115は、ステップS703で分割された新たな論理識別子範囲Range0（論理識別子範囲[8, 9]）がNullであるかどうかを判定する（ステップS704）。上記の例では、論理識別子範囲Range0がNullではないので（ステップS704のNo）、CPU115は、ノード406の0側次ノードであるノード412にメッセージを転送して、そのノード412において論理識別子範囲Range0を検索させる（ステップS705）。

【 0 1 1 6 】

また、ノード406のCPU115は、ステップS703で分割された新たな論理識別子範囲Range1（論理識別子範囲[10, 11]）がNullであるかどうかを判定する（ステップS706）。この場合も、上記の例では、論理識別子範囲Range1がNullではないので（ステップS706のNo）、CPU115は、ノード406の1側次ノードであるノード413にメッセージを転送して、そのノード413において論理識別子範囲Range1を検索させる（ステップS707）。

【 0 1 1 7 】

以降は同様に、論理識別子範囲[8, 9]はノード412にて論理識別子範囲[8, 8]（図5の1212）と論理識別子範囲[9, 9]（図5の1211）に分割されて、それぞれノード408とノード409に転送される。論理識別子範囲[10, 11]もノード413にて論理識別子範囲[10, 10]（図5の1210）と論理識別子範囲[11, 11]（図5の1209）に分割され、それぞれノード410とノード411に転送される。

【 0 1 1 8 】

一方、ノード405に転送された論理識別子範囲[7, 7]は、上限値と下限値の先頭ビットは常に同一なので（ステップS602のYes）、論理識別子範囲分割はされず、論理識別子範囲が図5の1205から図5の1208、図5の1208から図5の1211

10

20

30

40

50

3 というように左ビットシフトを受けながら、ノード405からノード411へ、ノード411からノード407へと転送される。

【0119】

ノード407、ノード408、ノード409、ノード410、ノード411に至った論理識別子範囲は、各ノードの論理識別子範囲のみを含むため（図2のステップS701のYes）、各ノードのCPU115は結果を初期に要求を出したノード413に返す（ステップS702）。

【0120】

以上に説明したように、この実施の形態1では、検索対象の論理識別子範囲に属する論理識別子を担うノードに、検索メッセージが行き渡るように、先頭ビットの違いによって論理識別子範囲を分割しながら、左ビットシフトを実行し、論理識別子範囲を分割する手法を用いることで、定数次数と対数ホップ数を実現しつつグラフ上での範囲検索を実現することができる。

【0121】

実施の形態2.

上記の実施の形態1はオーバレイネットワークに関係なく、de Bruijnグラフを用いた理論上の範囲検索のアルゴリズムについての説明であったが、実際には、Chordのハッシュ空間では各ノードは間隔を持って存在しているため（すなわち、例えば論理識別子空間16（2の4乗）である場合に、論理識別子0～15の全てにノードが存在するわけではないため）、全ての論理識別子を持つノードが定義されたde Bruijnグラフは適用できない。そこで、この実施の形態2ではde Bruijn上の論理識別子（ID）をChordハッシュ空間上の仮想IDとし、一つのノードがある区間のIDを担当させている。

【0122】

図6は、この実施の形態2におけるデータ検索システムの構成を示すブロック図である。図6に示すように、この実施の形態2におけるデータ検索システムは、ネットワーク800上でのアドレスを有する複数のピア810、820、830から構成される。図6には、ネットワーク800に接続されているピアはピア810、820、830だけであるが、ほかにも複数のピアがネットワーク800に接続されている。各ピアは、プログラム制御により動作するサーバなどのコンピュータ（中央処理装置；プロセッサ；データ処理装置）で構成されている（コンピュータのハードウェア構成は図1に示した構成と同様である）。なお、この実施の形態において、ピアとノードとは同義としている。

【0123】

図1に示すように、ピア810は、ピア論理識別子格納部818、ローカルデータ格納部811、ルーティングテーブル812、メッセージ転送手段813、通信手段814、範囲分割手段815、左ビットシフト範囲分割手段816、登録・検索実行手段817を備えている。なお、ピア810以外のピア820、830も、ピア810と同一の構成を備えている。

【0124】

ピア810を構成する格納部（ローカルデータ格納部811、ルーティングテーブル812、ピア論理識別子格納部818）には、それぞれ概略つぎのようなデータが格納されている。

【0125】

ピア論理識別子格納部818には、オーバレイネットワークでの当該ピア（例えばピア810）の論理識別子（ノードIDともいう）が格納されている。論理識別子はオーバレイネットワーク上における各ピアを識別するための識別子である。

【0126】

ローカルデータ格納部811には、全ピアで共有するデータ（コンテンツ）のうち当該ピア（例えばピア810）が担当するデータが格納される。各ピアがどの部分のデータを担当（管理）するかは、ピア論理識別子格納部818に格納された各ピアの論理識別子に

10

20

30

40

50

よって異なる。

【0127】

ルーティングテーブル812は、各ピアがルーティングを行うときに参照するテーブルである。ルーティングテーブル812には、他のピアの論理識別子と、ネットワーク800上でのアドレス（例えば、Internet Protocol (IP) アドレス）の組が複数格納されている。このルーティングテーブルの管理方式（すなわち、全ピアの中から、どのピアの論理識別子とネットワークアドレスの組を持つか）は、非特許文献1に記載のKoordeと同一である。上述したように、ルーティングテーブル812には、当該ピアの論理識別子の2倍のpredecessorの論理識別子とアドレスの組、および当該ピアのsuccessorの論理識別子とアドレスの組が格納される。

10

【0128】

ピア810を構成する手段（メッセージ転送手段813、通信手段814、範囲分割手段815、左ビットシフト範囲分割手段816、登録・検索実行手段817）は、それぞれ概略つぎのように動作する。

【0129】

範囲分割手段815は、与えられた論理識別子範囲を、ルーティングテーブル812に格納されたピアの論理識別子を境界値として分割する。この場合、例えば、論理識別子範囲にピアの論理識別子が含まれる場合には、最小論理識別子とピアの論理識別子とから決定される論理識別子範囲と、ピアの論理識別子と最大論理識別子とから決定される論理識別子範囲とに分割される。このとき、シフト後最大論理識別子やシフト後最小論理識別子範囲も同時に分割する。

20

【0130】

左ビットシフト範囲分割手段816は、与えられた論理識別子範囲について左ビットシフトを行うとともに、左ビットシフト前の先頭ビットによって異なる複数の論理識別子範囲を出力する。この手順は、図3に示した実施の形態1における左ビットシフト範囲分割処理の手順と同一である。

【0131】

登録・検索実行手段817は、データの登録および検索の処理を実行する。例えば、登録・検索実行手段817は、外部プログラムから検索要求があった場合に呼ばれ、検索条件に応じて論理識別子範囲を生成し、これをメッセージ転送手段に渡し、返ってきた検索結果を外部プログラムに与える役割を担う。

30

【0132】

メッセージ転送手段813は、データの登録要求および検索メッセージ（検索要求）の転送処理を実行する。メッセージ転送先の決定には、メッセージ内に含まれる論理識別子範囲が用いられる。この論理識別子範囲には、最大論理識別子と最小論理識別子、およびシフト後最大論理識別子、シフト後最小論理識別子が格納されている。

【0133】

通信手段814は、他ピアのネットワーク800上でのアドレスを指定して、ネットワーク800上にメッセージを送受信する処理を担う。これにより、あるピアのメッセージ転送手段813によって他のピアのメッセージ転送手段813を呼び出すことが可能となる。

40

【0134】

次に、この実施の形態2におけるデータ検索システムの動作について説明する。

【0135】

図7は、この実施の形態2のデータ検索システムの全体動作を示すフローチャートである。図7に示す処理において、まず、ピアmにて登録・実行処理部817が、検索式Queryを含んだ検索要求を受け付け、検索式Queryに対応する論理識別子範囲リストRangesに変換して取得する（ステップS1001）。なお、検索要求は、例えば検索要求者（ユーザ）の入力部（図1の116）の操作に応じてピアmの登録・検索実行手段817に入力される。

50

【0136】

次に、ピアmの登録・検索実行手段817は、検索結果を格納するリストResultsを初期化し(ステップS1002)、他ピアからの検索結果メッセージ(受信結果)をこのリストResultsに格納するプロセスの起動を行う(ステップS1002)。

【0137】

次に、ピアmの登録・検索実行手段817は、検索結果を待つ最大時間をタイムアウト(Timeout)としてタイマに設定する(ステップS1003)。そして、ピアmの登録・検索実行手段817は、仮想ノードiを設定する(ステップS1004)。好適には、この仮想ノードiの論理識別子はピアmとsuccessorのピアの間から任意に選ばれる。

10

【0138】

次いで、ピアmの登録・検索実行手段817は、ステップS1001で得られた論理識別子範囲リストと仮想ノードiを引数として、メッセージ転送手段813を非同期に呼び出す(ステップS1005)。これにより、メッセージ転送手段813において、図8に示すフローチャートの各処理が実行される。

【0139】

その後、ピアmの登録・検索実行手段817は、論理子範囲リスト内の全ての論理識別子に対する検索結果が得られたか、あるいはTimeoutが0以下となるまで、待機し(待ち続け)(ステップS1006)、一定時間(Interval)毎にタイマの値を減算していく(ステップS1007)。

20

【0140】

待ち条件を抜けた後(論理識別子範囲リスト内の全ての論理識別子に対する検索結果が得られたとき、あるいはタイマがタイムアウトしたとき)は、ピアmの登録・検索実行手段817は、他ピアから送信された検索結果が格納されている結果リストResultsを、検索要求者に対して提供する(ステップS1008)。

【0141】

図8は、この実施の形態2のデータ検索手順を示すフローチャートである。ピアmにおいて、メッセージ転送手段813は、論理識別子範囲idRangesを仮想ノードiで探す要求を受けると(図7のステップS1005)、論理識別子範囲idRangesをSuccessorの論理識別子sとともに範囲分割手段815に渡す。範囲分割手段815は、論理識別子範囲idRangesをSuccessorの論理識別子sで2つの論理識別子範囲LowRangesとHighRangesに分割する(ステップS901)。そして、範囲分割手段815は、分割した2つの論理識別子範囲LowRangesとHighRangesをメッセージ転送手段813に渡す。

30

【0142】

メッセージ転送手段813は、論理識別子範囲LowRangesがNullであるか否か判定する(ステップS902)。論理識別子範囲LowRangesがNullでない場合には(ステップS902のNo)、メッセージ転送手段813は、ローカルデータ格納部811において、論理識別子範囲LowRangeに含まれるシフト前の論理識別子をもとに、これと対応するデータ(論理識別子範囲LowRangeに対応する検索式に合致するデータ)を検索する(ステップS903)。そして、検索した検索結果を検索実行者に渡す。

40

【0143】

次いで、メッセージ転送手段813は、論理識別子範囲HighRangesがNullであるか否か判定する(ステップS904)。論理識別子範囲HighRangeがNullである場合には(ステップS904のYes)、このピアmでの検索処理の実行を終了する。論理識別子範囲HighRangesがNullでない場合には(ステップS904のNo)、メッセージ転送手段813は、仮想ノードiが(m, s]に含まれるか否かを判定する(ステップS905)。(m, s]は、m(ピアmの論理識別子)よりも大きくs(Successorの論理識別子)以下であることを示している。

50

【0144】

仮想ノード i が $(m, s]$ の範囲に含まれない場合は (ステップ S905 の No)、メッセージ転送手段 813 は、ピア s (Successor) に同一の論理識別子範囲 $idRanges$ と仮想ノード i を引数としてメッセージ転送手段 813 を呼び出す (ステップ S911)。これにより、ピア s (Successor) において、論理識別子範囲 $idRanges$ を仮想ノード i で探す処理 (図 8 に示す処理) が実行される。そして、このピア m での検索処理の実行を終了する。

【0145】

仮想ノード i が $(m, s]$ の範囲に含まれる場合は (ステップ S905 の Yes)、メッセージ転送手段 813 は、左ビットシフト範囲分割手段 816 に論理識別子範囲 $HighRanges$ を渡す。左ビットシフト範囲分割手段 816 は、左ビットシフトを行い、論理識別子範囲 $HighRanges$ を 2 つの論理識別子範囲 $Range0$ と $Range1$ に分割する処理を実行する (ステップ S906)。なお、左ビットシフト範囲分割手段 816 による論理識別子範囲を分割する手順は、図 3 に示した実施の形態 1 における左ビットシフト範囲分割処理の手順と同一である。左ビットシフト範囲分割手段 816 は、分割した 2 つの論理識別子範囲 $Range0$ 、 $Range1$ をメッセージ転送手段 813 に渡す。

【0146】

次いで、メッセージ転送手段 813 は、論理識別子範囲 $Range0$ が Null であるか否か判定する (ステップ S907)。論理識別子範囲 $Range0$ が Null でない場合には (ステップ S907 の No)、メッセージ転送手段 813 は、ピア d (Predecessor) のメッセージ転送手段 813 を、論理識別子範囲 $Range0$ 、仮想ノード $2 \times i \pmod{N}$ 、すなわち仮想ノード i を左ビットシフトして得られる値で呼び出す (ステップ S908)。これにより、ピア d (Predecessor) において、論理識別子範囲 $Range0$ を仮想ノード $2 \times i \pmod{N}$ で探す処理 (図 8 に示す処理) が実行される。

【0147】

さらに、メッセージ転送手段 813 は、論理識別子範囲 $Range1$ が Null であるか否か判定する (ステップ S909)。論理識別子範囲 $Range1$ が Null でない場合には (ステップ S909 の No)、メッセージ転送手段 813 は、ピア d のメッセージ転送手段 813 を、論理識別子範囲 $Range1$ 、仮想ノード $2 \times i + 1 \pmod{N}$ 、すなわち仮想ノード i を左ビットシフトして、最下位ビットを 1 とした値で呼び出す (ステップ S910)。これにより、ピア d (Predecessor) において、論理識別子範囲 $Range1$ を仮想ノード $2 \times i + 1 \pmod{N}$ で探す処理 (図 8 に示す処理) が実行される。そして、処理を終了する。

【0148】

次に、この実施の形態 2 におけるデータ検索システムの動作の具体例 (データ検索処理の手順の具体例) について、図 9 および図 10 を用いて説明する。

【0149】

図 9 は、この実施の形態 2 におけるピア間でのメッセージの流れの具体例を示す説明図である。また、図 10 は、この実施の形態 2 における左ビットシフト範囲分割処理のアルゴリズムを視覚的に示す説明図である。

【0150】

図 9 において、オーバーレイネットワーク上に参加している複数のピアを示す。論理識別子 13 であるピア 1316 が、検索式 $Query$ として {7 温度 11} を実行するものとし、温度という属性は論理識別子範囲と対応するものとする。

【0151】

この場合、ピア 1316 の登録・検索実行手段 817 は、検索式 $Query$ から対応する論理識別子範囲 $idRanges$ を算出し、論理識別子範囲 [7, 11] が得られる (ステップ S1001)。次に、登録・検索実行手段 817 は、結果リスト $Results$

10

20

30

40

50

を初期化して、他ピアからの結果を受け付けるプロセスを立ち上げ（ステップS1002）、タイマにタイムアウトを設定した後に（ステップS1003）、仮想ノードiを設定する（ステップS1004）。

【0152】

仮想ノードiは任意の値でもよいが、好適には自身の論理識別子よりも大きくSuccessorの論理識別子以下の論理識別子であり、かつ、その下位ビットに宛先の論理識別子の先頭ビットをより長く含める。この場合、論理識別子1110と論理識別子1111が前者の条件を満たすため、この中で、可能な限り長く、宛先の論理識別子の先頭ビットを下位ビットに持つように左ビットシフト範囲分割を行ってよい。

【0153】

図10に示す論理識別子範囲1401は、左ビットシフト範囲分割（図3に示した処理）により、[8, 11]である論理識別子範囲1402と[7, 7]である論理識別子範囲1403に分割される。前者の論理識別子範囲[8, 11]のシフト前先頭ビットは1であり、後者の論理識別子範囲[7, 7]のシフト前先頭ビットは0である。さらにピア1316にて左ビットシフト範囲分割をすることも可能であるが、後者の論理識別子範囲[7, 7]をさらに左ビットシフト範囲分割すると、上記条件である論理識別子1110と論理識別子1111の下位から2ビット目に0（シフト前先頭ビット）であるものは存在しないので、さらなる左ビットシフト範囲分割は行わない。一方、前者の論理識別子範囲[8, 11]はさらにピア1316にて左ビットシフト分割を行うことが可能である。

【0154】

図10に示す論理識別子範囲1402（論理識別子範囲[8, 11]）に対して左ビットシフト範囲分割を行うと、先頭ビットとして0が得られ、論理識別子範囲1404（論理識別子範囲[8, 11]）が得られる。ここで、さらなる左ビットシフト範囲分割は不可能であるため、これ以上の分割は行わない。

【0155】

以上から、ピア1316におけるステップS1005では、論理識別子範囲[8, 11]を仮想ノード「1110」で探し（図9の1321）、さらに論理識別子範囲[7, 7]を仮想ノード「1110」で探す（図9の1322）。

【0156】

ピア1316において、登録・検索実行手段817は、メッセージ転送手段813を呼び出して、論理識別子範囲[8, 11]を仮想ノード「1110」で探す（ステップS1005）。これにより、メッセージ転送手段813によって図8に示した処理が実行される。

【0157】

ピア1316のメッセージ転送手段813は、まず、論理識別子範囲[8, 11]をピア1316のSuccessorであるピア1317の論理識別子15（「1111」）とともに範囲分割手段815に渡す。範囲分割手段815は、ピア1316のSuccessorであるピア1317の論理識別子15を境界値として、論理識別子範囲[8, 11]の分割を行うが（ステップS901）、この論理識別子範囲[8, 11]に論理識別子「15」は含まれないため、ここでは論理識別子範囲LowRangesにNullを入力し、論理識別子範囲HighRangesに[8, 11]を入力する。そのため、ステップS902の判定結果がYesとなり、ステップS904の判定結果がNoとなる。そして、メッセージ転送手段813は、仮想ノード「1110」が（1101, 1111）の範囲に含まれるか否かの判定を行う（ステップS905）。

【0158】

仮想ノード「1110」は（1101, 1111）の範囲に含まれるので（ステップS905のYes）、メッセージ転送手段813は、左ビットシフト範囲分割手段816に対して、当該論理識別子範囲[8, 11]を左ビットシフトして、2つの論理識別子範囲Range0とRange1に分割する処理を実行させる（ステップS906）。この結果、Range0には論理識別子範囲[8, 9]（図10の1407）が格納され、Ra

10

20

30

40

50

ng e 1には論理識別子範囲[1 0 , 1 1] (図 1 0 の 1 4 0 6) が格納される。

【 0 1 5 9 】

そして、メッセージ転送手段 8 1 3 は、論理識別子範囲 R a n g e 0 は N u l l である
か否か判定する(ステップ S 9 0 7)。このとき、論理識別子範囲 R a n g e 0 ([8 ,
9]) は N u l l でないので(ステップ S 9 0 7 の N o)、メッセージ転送手段 8 1 3 は
、ピア 1 3 1 6 の d ノード (P r e d e c e s s o r) であるピア 1 3 1 5 のメッセージ
転送手段 8 1 3 を、論理識別子範囲 [8 , 9]、仮想ノード「 1 1 0 0 」で呼び出す(ス
テップ S 9 0 8)。これは、図 9 の 1 3 2 3 に対応する。これにより、ピア 1 3 1 5 にお
いて、論理識別子範囲 [8 , 9] を仮想ノード「 1 1 0 0 」で探す処理(図 8 に示す処理
)が実行される。

10

【 0 1 6 0 】

また、メッセージ転送手段 8 1 3 は、論理識別子範囲 R a n g e 1 は N u l l であるか
否か判定する(ステップ S 9 0 9)。このとき、論理識別子範囲 R a n g e 1 ([1 0 ,
1 1]) は N u l l でないので(ステップ S 9 0 9 の N o)、メッセージ転送手段 8 1 3
は、ピア 1 3 1 6 の d ノード (P r e d e c e s s o r) であるピア 1 3 1 5 のメッセ
ージ転送手段 8 1 3 を、論理識別子範囲 [1 0 , 1 1]、仮想ノード「 1 1 0 1 」で呼び出
す(ステップ S 9 1 0)。これは、図 9 の 1 3 2 4 に対応する。これにより、ピア 1 3 1
5 において、論理識別子範囲 [1 0 , 1 1] を仮想ノード「 1 1 0 1 」で探す処理(図 8
に示す処理)が実行される。

【 0 1 6 1 】

20

また、ピア 1 3 1 6 において、論理識別子範囲 [7 , 7] を仮想ノード「 1 1 1 0 」で
探す場合も同様に、登録・検索実行手段 8 1 7 は、メッセージ転送手段 8 1 3 を呼び出し
て、図 8 に示した処理を実行させる(ステップ S 1 0 0 5)。

【 0 1 6 2 】

ピア 1 3 1 6 のメッセージ転送手段 8 1 3 は、まず、論理識別子範囲 [7 , 7] をピア
1 3 1 6 の S u c c e s s o r であるピア 1 3 1 7 の論理識別子 1 5 (「 1 1 1 1 」)と
ともに範囲分割手段 8 1 5 に渡す。範囲分割手段 8 1 5 は、ピア 1 3 1 6 の S u c c e s
s o r であるピア 1 3 1 7 の論理識別子 1 5 を境界値として、論理識別子範囲 [7 , 7]
の分割を行うが(ステップ S 9 0 1)、この論理識別子範囲 [7 , 7] に論理識別子「 1
5 」は含まれないため、ここでは論理識別子範囲 L o w R a n g e s に N u l l を入力し
、論理識別子範囲 H i g h R a n g e s に [7 , 7] を入力する。そのため、ステップ S
9 0 2 の判定結果が Y e s となり、ステップ S 9 0 4 の判定結果が N o となる。そして、
メッセージ転送手段 8 1 3 は、仮想ノード「 1 1 1 0 」が (1 1 0 1 , 1 1 1 1] の範囲
に含まれるか否かの判定を行う(ステップ S 9 0 5)。

30

【 0 1 6 3 】

仮想ノード「 1 1 1 0 」は (1 1 0 1 , 1 1 1 1] の範囲に含まれるので(ステップ S
9 0 5 の Y e s)、メッセージ転送手段 8 1 3 は、左ビットシフト範囲分割手段 8 1 6 に
対して、当該論理識別子範囲 [7 , 7] を左ビットシフトして、2つの論理識別子範囲 R
a n g e 0 と R a n g e 1 に分割する処理を実行させる(ステップ S 9 0 6)。ここでは
、分割されずに左ビットシフトのみ行われ(図 3 のステップ S 6 0 2 , S 6 0 6 , S 6 0
7 参照)、R a n g e 0 に N u l l が格納され、R a n g e 1 に論理識別子範囲 [7 , 7
(図 1 0 の 1 4 0 5) が格納される。

40

【 0 1 6 4 】

そして、メッセージ転送手段 8 1 3 は、論理識別子範囲 R a n g e 0 は N u l l である
か否か判定する(ステップ S 9 0 7)。このとき、論理識別子範囲 R a n g e 0 は N u l l
であるので(ステップ S 9 0 7 の Y e s)、ステップ S 9 0 8 の処理を実行せずに、メ
ッセージ転送手段 8 1 3 は、論理識別子範囲 R a n g e 1 は N u l l であるか否か判定す
る(ステップ S 9 0 9)。このとき、論理識別子範囲 R a n g e 1 ([7 , 7]) は N u
l l でないので(ステップ S 9 0 9 の N o)、メッセージ転送手段 8 1 3 は、ピア 1 3 1
6 の d ノード (P r e d e c e s s o r) であるピア 1 3 1 5 のメッセージ転送手段 8 1

50

3を、論理識別子範囲[7, 7]、仮想ノード「1101」で呼び出す(ステップS910)。これは、図9の1325に対応する。これにより、ピア1315において、論理識別子範囲[7, 7]を仮想ノード「1101」で探す処理(図8に示す処理)が実行される。

【0165】

ピア1315において、論理識別子範囲[8, 9](図10の1407)を仮想ノード「1100」で探す処理、論理識別子範囲[10, 11](図10の1406)を仮想ノード「1101」で探す処理、および論理識別子範囲[7, 7](図10の1405)を仮想ノード「1101」で探す処理が行われる。

【0166】

これらの処理のいずれにおいても、まず、ピア1315の範囲分割手段815が、ピア1315のSuccessorの論理識別子13(「1101」)を境界値として分割する(ステップS901)。ここでは、論理識別子範囲[10, 11]だけが、論理識別子範囲[10, 10](図10の1410)と論理識別子範囲[11, 11](図10の1409)に分割される。そして、ステップS903において、ピア1315のメッセージ転送手段813がSuccessorであるピア1316にこの範囲[11, 11]のデータをローカルデータ格納部811から取得するように要求する。

【0167】

この分割で残った論理識別子範囲[7, 7](図10の1405)、論理識別子範囲[8, 9](図10の1407)、論理識別子範囲[10, 10](図10の1410)は、それぞれ、左ビットシフト範囲分割手段816によって左ビット範囲分割が行われる。

【0168】

すなわち、論理識別子範囲[7, 7](図10の1405)は論理識別子範囲[7, 7](図10の1408)に変換され、論理識別子範囲[8, 9](図10の1407)は論理識別子範囲[8, 8](図10の1412)と論理識別子範囲[9, 9](図10の1411)に分割され、論理識別子範囲[10, 10](図10の1410)は論理識別子範囲[10, 10](図10の1413)に変換される。

【0169】

これらの論理識別子範囲は、ステップS908あるいはステップS910において、ピア1315のdノード(predecessor)であるピア1312に転送される。すなわち、仮想ノード「1011」での論理識別子範囲[7, 7]の検索を要求するメッセージがピア1312に転送され(図9の1329)、仮想ノード「1000」での論理識別子範囲[8, 8]の検索を要求するメッセージがピア1312に転送され、仮想ノード「1001」での論理識別子範囲[9, 9]の検索を要求するメッセージがピア1312に転送され、仮想ノード「1010」での論理識別子範囲「10, 10」の検索を要求するメッセージがピア1312に転送される。

【0170】

ピア1312では、いずれの範囲もピア1312の論理識別子3(2進数表示で「0011」)とそのSuccessorの論理識別子6(2進数表示で「0110」)には含まれず、同様に、仮想ノードも含まれないため(ステップS905のNo)、ステップS911にて、全てSuccessorであるピア1313に転送される(図9の1330~1333)。

【0171】

ピア1313では、当該ピア1313の論理識別子6(2進数表示で「0110」)と、このピア1313のSuccessorであるピア1314の論理識別子9(2進数表示で「1001」)との間に、論理識別子範囲[7, 7](図10の1333)、論理識別子範囲[8, 8](図10の1332)、論理識別子範囲[9, 9](図10の1331)が含まれるため、ピア1313はピア1314に対してこれらの範囲をローカルデータ格納部811から探して、検索実行者であるピア1316に返すように指示する(ステップS903)。残った範囲である[10, 10](図10の1330)については、そ

10

20

30

40

50

の仮想ノードが *Successor* との間に存在しないため、ステップ *S 9 1 1* にてピア *1 3 1 3* は *Successor* であるピア *1 3 1 4* に論理識別子範囲 $[10, 10]$ を転送する (図 9 の *1 3 3 4*)。

【0172】

ピア *1 3 1 4* では、当該ピア *1 3 1 4* の論理識別子 *9* (2進数表示で「*1 0 0 1*」) と、このピア *1 3 1 4* の *Successor* であるピア *1 3 1 5* の論理識別子 *1 0* (2進数表示で「*1 0 1 0*」) との間に、論理識別子範囲 $[10, 10]$ (図 10 の *1 3 3 4*) が含まれるため、ピア *1 3 1 4* はピア *1 3 1 5* に対してこの範囲 $[10, 10]$ をローカルデータ格納部 *8 1 1* から探して、検索実行者であるピア *1 3 1 6* に返すように指示する (ステップ *S 9 0 3*)。

10

【0173】

以上により、全ての論理識別子範囲 *idRanges* ($[7, 11]$) の検索結果がピア *1 3 1 6* に返される (ステップ *S 1 0 0 6*)。そして、ピア *1 3 1 6* の登録・検索実行手段 *8 1 7* は、他ピアから送信された検索結果が格納されている結果リスト *Results* を検索要求者に対して提供する (ステップ *S 1 0 0 8*)。

【0174】

以上に説明したように、この実施の形態 2 では、検索式に合致する集合を論理識別子空間における範囲に変換し、この論理識別子範囲に属する論理識別子を担うピアに、検索メッセージが行き渡るように、論理識別子範囲とともに検索メッセージを分割する。このとき、論理識別子範囲に対する左ビットシフトを実行し、先頭ビットの違いによって、論理識別子範囲を分割する。このような手法を用いることで、*de Bruijn* グラフ上での範囲検索と同様に、定数次数と対数ホップ数を実現しつつオーバーレイネットワーク上での範囲検索を実現することができる。

20

【0175】

実施の形態 3 .

上記の実施の形態 2 は、*de Bruijn* 上の論理識別子 (*ID*) を *Chord* ハッシュ空間上の仮想 *ID* とし、論理識別子範囲の先頭ビットの違いにより、新たな仮想 *ID* を生成しつつ論理識別子範囲を分割していた。しかし、この場合、大きなハッシュ空間 (*ID* 空間) を用いる場合には、ビット数が多くなり、管理する仮想 *ID* の数も多くなる。そこで、この実施の形態 3 では、複数の仮想 *ID* を集合として、すなわち仮想範囲 (仮想論理識別子範囲) として管理することにより、ピアが扱うデータ量を削減する。

30

【0176】

図 11 は、この実施の形態 3 におけるデータ検索システムの構成を示すブロック図である。図 11 に示すように、この実施の形態 3 におけるデータ検索システムは、ネットワーク *2 0 0 0* 上でのアドレスを有する複数のピア *2 0 1 0* , *2 0 2 0* , *2 0 3 0* から構成される。図 11 には、ネットワーク *2 0 0 0* に接続されているピアはピア *2 0 1 0* , *2 0 2 0* , *2 0 3 0* だけであるが、ほかにも複数のピアがネットワーク *2 0 0 0* に接続されている。各ピアは、プログラム制御により動作するサーバなどのコンピュータ (中央処理装置; プロセッサ; データ処理装置) で構成されている (コンピュータのハードウェア構成は図 1 に示した構成と同様である)。なお、この実施の形態において、ピアとノードとは同義としている。

40

【0177】

図 11 に示すように、ピア *2 0 1 0* は、ピア論理識別子格納部 *2 0 1 8*、ローカルデータ格納部 *2 0 1 1*、ルーティングテーブル *2 0 1 2*、メッセージ転送手段 *2 0 1 3*、通信手段 *2 0 1 4*、検索範囲左ビットシフト手段 *2 0 1 5*、範囲分割手段 *A 2 0 1 6*、範囲分割手段 *B 2 0 1 7*、登録・検索実行手段 *2 0 1 9* を備えている。なお、ピア *2 0 1 0* 以外のピア *2 0 2 0* , *2 0 3 0* も、ピア *2 0 1 0* と同一の構成を備えている。

【0178】

ピア *2 0 1 0* を構成する格納部 (ローカルデータ格納部 *2 0 1 1*、ルーティングテーブル *2 0 1 2*、ピア論理識別子格納部 *2 0 1 8*) は、上記の実施の形態 2 と同じ構成である

50

ため、説明を省略する。

【0179】

また、ピア2010を構成する通信手段2014、および登録・検索実行手段2019も、上記の実施の形態2と同じ構成であるため、説明を省略する。

【0180】

ピア2010を構成する検索範囲左ビットシフト手段2015、範囲分割手段A2016、および範囲分割手段B2017は、それぞれ概略つぎのように動作する。

【0181】

これらの手段では、検索対象を検索範囲として定義する。検索範囲には、仮想範囲と、論理識別子範囲と、シフト回数とが含まれる（図166の1920参照）。このようにして、この実施の形態3では、実施の形態2において個別に扱っていた仮想ノードを集合として扱い、扱うデータ量を削減する。

10

【0182】

検索範囲左ビットシフト手段2015は、シフト回数を増加させ、仮想範囲の上限値および下限値を更新する処理を実行する。仮想範囲の上限値は、仮想範囲の上限値を左ビットシフトし、下位ビットに論理識別子範囲の上限値あるいは下限値の先頭ビットを取り出して割り当てる。なお、検索範囲左ビットシフト手段2015が実行する処理の詳細については図15を参照して後述する。

【0183】

範囲分割手段A2016は、検索範囲内の仮想範囲を、与えられた範囲に属する否かで分割する。その際、検索範囲内の論理識別子範囲についても分割する。なお、範囲分割手段A2016が実行する処理の詳細については図14を参照して後述する。

20

【0184】

範囲分割手段B2017は、検索範囲内の論理識別子範囲を、与えられた範囲に属する否かで分割する。その際、検索範囲内の仮想識別子範囲についても分割する。なお、範囲分割手段B2017が実行する処理の詳細については図13を参照して後述する。

【0185】

次に、この実施の形態3におけるデータ検索システムの動作について説明する。

【0186】

この実施の形態3のデータ検索システムの全体動作は、上記の実施の形態2のデータ検索システムの全体動作（図7）と略同様である。ただし、この実施の形態3では、ピアmにて論理識別子範囲を探す（検索する）方式が異なる。すなわち、この実施の形態3では、図7のステップS1004における「ピアmにて論理識別子範囲idRangesを仮想ノードiで探す」処理の代わりに、図12に示す「ピアmにて検索範囲集合qRangesを探す」処理が行われる。なお、この実施の形態3では、図7における仮想ノードの初期化（図7のステップS1003）の代わりに「検索範囲（仮想範囲と論理識別子範囲を含む）の初期化」が行われる。

30

【0187】

図12は、この実施の形態3のデータ検索手順を示すフローチャートである。ピアmにおいて、メッセージ転送手段2013は、登録・検索実行手段2019からの検索範囲集合qRangesを探す要求を受けると、発見検索範囲集合foundRanges、未発見検索範囲集合NotFoundRanges、ピアd転送検索範囲集合dRanges、ピアs転送検索範囲集合sRangesを初期化する（ステップS1601）。

40

【0188】

そして、メッセージ転送手段2013は、検索範囲集合qRanges内の各検索範囲qRangeを、ピアmとピアs（Successor）の論理識別子の範囲（m,s]とともに範囲分割手段B2017に渡し、各検索範囲qRangeに対して、ステップS1603の処理を繰り返し実行させる（ステップS1602）。

【0189】

範囲分割手段B2017は、検索範囲qRangeを、ピアmとピアsの論理識別子の

50

範囲 $(m, s]$ で、論理識別子範囲をもとに範囲内検索範囲 $inRange$ と範囲外検索範囲 $outRange$ に分割する（ステップ 1603）。そして、範囲分割手段 B2017 は、範囲内検索範囲 $inRange$ と範囲外検索範囲 $outRange$ をメッセージ転送手段 2013 に返す。

【0190】

メッセージ転送手段 2013 は、範囲分割手段 B2017 から各検索範囲 $qRange$ に対する各範囲内検索範囲 $inRange$ を順次取得し、範囲分割手段 B2017 から各検索範囲 $qRange$ に対する各範囲外検索範囲 $outRange$ を順次取得する。各々の範囲内検索範囲 $inRange$ の集合が範囲内検索範囲集合 $inRangeSet$ であり、各々の範囲外検索範囲 $outRange$ の集合が範囲外検索範囲集合 $outRanges$ である。

10

【0191】

メッセージ転送手段 2013 は、検索範囲集合 $qRanges$ 内の検索範囲 $qRange$ に対して得られた範囲内検索範囲集合 $inRangeSet$ と範囲外検索範囲集合 $outRanges$ を、それぞれ、発見検索範囲集合 $foundRanges$ と未発見検索範囲集合 $NotFoundRanges$ に追加（格納）する（ステップ S1604）。

【0192】

次いで、メッセージ転送手段 2013 は、未発見検索範囲集合 $NotFoundRanges$ 内の未発見検索範囲 $NotFoundRange$ を、ピア m とピア s の論理識別子の範囲 $(m, s]$ とともに範囲分割手段 A2016 に渡し、各未発見検索範囲 $NotFoundRange$ に対して、ステップ S1606 の処理を繰り返し実行させる（ステップ S1605）。

20

【0193】

範囲分割手段 A2016 は、未発見検索範囲 $NotFoundRange$ を、ピア m とピア s の論理識別子の範囲 $(m, s]$ で、仮想範囲をもとに範囲内検索範囲 $inRange$ と範囲外検索範囲 $outRange$ に分割する（ステップ S1606）。そして、範囲分割手段 A2016 は、範囲内検索範囲 $inRange$ と範囲外検索範囲 $outRange$ をメッセージ転送手段 2013 に返す。

【0194】

メッセージ転送手段 2013 は、範囲分割手段 A2016 から各未発見検索範囲 $NotFoundRange$ に対する各範囲内検索範囲 $inRange$ を順次取得し、範囲分割手段 A2016 から各未発見検索範囲 $NotFoundRange$ に対する各範囲外検索範囲 $outRange$ を順次取得する。各々の範囲内検索範囲 $inRange$ の集合が範囲内検索範囲集合 $inRanges$ であり、各々の範囲外検索範囲 $outRange$ の集合が範囲外検索範囲集合 $outRanges$ である。

30

【0195】

次いで、メッセージ転送手段 2013 は、範囲内検索範囲集合 $inRanges$ 内の要素 $inRange$ を検索範囲左ビットシフト手段 2015 に順次渡し、ステップ S1608 の処理を繰り返し実行させる（ステップ S1607）。

【0196】

検索範囲左ビットシフト手段 2015 は、範囲内検索範囲 $inRange$ に対して、左ビットシフトの処理を実行する（ステップ S1608）。そして、左ビットシフト後の範囲内検索範囲をメッセージ転送手段 2013 に返す。

40

【0197】

メッセージ転送手段 2013 は、範囲内検索範囲集合 $inRanges$ と範囲外検索範囲集合 $outRanges$ を、それぞれ、ピア d 転送検索範囲集合 $dRanges$ とピア s 転送検索範囲集合 $sRanges$ に追加（格納）する（ステップ S1609）。

【0198】

次いで、メッセージ転送手段 2013 は、発見検索範囲集合 $foundRanges$ の要素数が 0 でなければ（ステップ S1610 の N）、ピア s にてローカルデータから範囲

50

`foundRanges`を探させる(ステップS1611)。

【0199】

また、メッセージ転送手段2013は、ピアd転送検索範囲集合`dRanges`の要素数が0でなければ(ステップS1612のN)、ピアdにて`dRanges`を探させる(ステップS1613)。同様に、ピアs転送検索範囲集合`sRanges`の要素数が0でなければ(ステップS1614のN)、ピアsにて`sRanges`を探させる(ステップS1615)。そして、処理を終了する。

【0200】

次に、範囲分割手段B2017が実行する処理(ステップS1603)について説明する。図13は、この実施の形態3の範囲分割手段Bが実行する処理を示すフローチャートである。なお、検索範囲`qRange`は、仮想検索範囲`iRange`と論理識別子範囲`kRange`とシフト回数`c`とから構成されている。

【0201】

範囲分割手段B2017は、まず、検索範囲`qRange`中で、範囲`Range`(自身の論理識別子と`Successor`の論理識別子の間の範囲)に含まれる範囲を範囲内仮想範囲集合`inKeyRanges`に格納し、含まれない範囲を範囲外仮想範囲集合`outKeyRanges`に格納する(ステップS1801)。そして、範囲分割手段B2017は、範囲内検索範囲集合`inQRanges`と範囲外検索範囲集合`outQRanges`を初期化する(ステップS1802)。

【0202】

次いで、範囲分割手段B2017は、範囲内仮想範囲集合`inKeyRanges`と範囲外仮想範囲集合`outKeyRanges`内の各仮想範囲`eachKeyRange`に対して、ステップS1804～S1809の処理を繰り返し実行する(ステップS1803)。

【0203】

範囲分割手段B2017は、仮想範囲`eachKeyRange`の上限値の先頭から(全ビット長 - シフト回数`c`)ビットを上位に、論理識別子範囲`kRange`の上限値を(全ビット長 - シフト回数`c`)回右ビットシフトしたビット列を下位に持つビット列を新たな仮想論理識別子範囲の上限値(最大仮想論理識別子)`NewImgRangeMax`とする(ステップS1804)。

【0204】

また、範囲分割手段B2017は、仮想範囲`eachKeyRange`の下限値の先頭から(全ビット長 - シフト回数`c`)ビットを上位に、論理識別子範囲`kRange`の上限値を(全ビット長 - シフト回数`c`)回右ビットシフトしたビット列を下位に持つビット列を新たな仮想論理識別子範囲の下限値(最小仮想論理識別子)`NewImgRangeMin`とする(ステップS1805)。

【0205】

そして、範囲分割手段B2017は、最大仮想論理識別子`NewImgRangeMax`と最小仮想論理識別子`NewImgRangeMin`から、新たな仮想論理識別子範囲`NewImgRange`を作成し、これと仮想範囲`eachKeyRange`とシフト回数`c`から、新たな検索範囲`NewQRange`を作成する(ステップS1806)。

【0206】

次いで、範囲分割手段B2017は、仮想範囲`eachKeyRange`が範囲内仮想範囲集合`inKeyRanges`に属するかどうかを判定する(ステップS1807)。仮想範囲`eachKeyRange`が範囲内仮想範囲集合`inKeyRanges`に属すると判定した場合は(ステップS1807のY)、範囲内検索範囲集合`inQRanges`に新たな検索範囲`NewQRange`を登録する(ステップS1808)。一方、仮想範囲`eachKeyRange`が範囲内仮想範囲集合`inKeyRanges`に属しないと判定した場合は(ステップS1807のN)、範囲外検索範囲集合`outKeyRanges`に新たな検索範囲`NewQRange`を登録する(ステップS1809)。

10

20

30

40

50

【 0 2 0 7 】

そして、範囲分割手段 B 2 0 1 7 は、範囲内検索範囲集合 $inQRanges$ と範囲外検索範囲集合 $outQRanges$ をメッセージ転送手段 2 0 1 3 に返す（ステップ S 1 8 1 0）。範囲内検索範囲集合 $inQRanges$ がメッセージ転送手段 2 0 1 3 によって範囲内検索範囲集合 $inRangeSet$ として取得され、範囲外検索範囲集合 $outQRanges$ がメッセージ転送手段 2 0 1 3 によって範囲外検索範囲集合 $outRanges$ として取得される。

【 0 2 0 8 】

次に、範囲分割手段 A 2 0 1 6 が実行する処理（ステップ S 1 6 0 6）について説明する。図 1 4 は、この実施の形態 3 の範囲分割手段 A が実行する処理を示すフローチャートである。なお、検索範囲 $qRange$ は、仮想検索範囲 $iRange$ と論理識別子範囲 $kRange$ とシフト回数 c とから構成されている。

10

【 0 2 0 9 】

範囲分割手段 A 2 0 1 6 は、まず、検索範囲 $qRange$ 中で、範囲 $Range$ （自身の論理識別子と $Successor$ の論理識別子の間の範囲）に含まれる範囲を範囲内仮想範囲集合 $inImgRanges$ に格納し、含まれない範囲を範囲外仮想範囲集合 $outImgRanges$ に格納する（ステップ S 1 7 0 1）。そして、範囲分割手段 A 2 0 1 6 は、範囲内検索範囲集合 $inQRanges$ と範囲外検索範囲集合 $outQRanges$ を初期化する（ステップ S 1 7 0 2）。

【 0 2 1 0 】

20

次いで、範囲分割手段 A 2 0 1 6 は、範囲内仮想範囲集合 $inImgRanges$ と範囲外仮想範囲集合 $outImgRanges$ 内の各仮想範囲 $eachImgRange$ に対して、ステップ S 1 7 0 4 ~ S 1 7 0 9 の処理を繰り返し実行する（ステップ S 1 7 0 3）。

【 0 2 1 1 】

範囲分割手段 A 2 0 1 6 は、仮想範囲 $eachImgRange$ の上限値を（全ビット長 - シフト回数 c ）回左ビットシフトし、下位の（全ビット長 - シフト回数 c ）ビットに 1 を並べたビット列と、論理識別子範囲 $kRange$ の上限値を比較し、小さい方を新たな論理識別子範囲の上限値（最大論理識別子） $NewRangeMax$ とする（ステップ S 1 7 0 4）。

30

【 0 2 1 2 】

また、範囲分割手段 A 2 0 1 6 は、仮想範囲 $eachImgRange$ の下限値を（全ビット長 - シフト回数 c ）回左ビットシフトし、下位の（全ビット長 - シフト回数 c ）ビットに 0 を並べたビット列と、論理識別子範囲 $kRange$ の下限値を比較し、大きい方を新たな論理識別子範囲の下限値（最小論理識別子） $NewRangeMin$ とする（ステップ S 1 7 0 5）。

【 0 2 1 3 】

そして、範囲分割手段 A 2 0 1 6 は、最大論理識別子 $NewRangeMax$ と最小論理識別子 $NewRangeMin$ から、新たな論理識別子範囲 $NewRange$ を作成し、これと仮想範囲 $eachImgRange$ とシフト回数 c から、新たな検索範囲 $NewQRanges$ を作成する（ステップ S 1 7 0 6）。

40

【 0 2 1 4 】

次いで、範囲分割手段 A 2 0 1 6 は、仮想範囲 $eachImgRange$ が範囲内仮想範囲集合 $inImgRanges$ に属するかどうかを判定する（ステップ S 1 7 0 7）。仮想範囲 $eachImgRange$ が範囲内仮想範囲集合 $inImgRanges$ に属すると判定した場合は（ステップ S 1 7 0 7 の Y）、範囲内検索範囲集合 $inQRanges$ に新たな検索範囲 $NewQRanges$ を登録する（ステップ S 1 7 0 8）。一方、仮想範囲 $eachImgRange$ が範囲内仮想範囲集合 $inImgRanges$ に属しないと判定した場合は（ステップ S 1 7 0 7 の N）、範囲外検索範囲集合 $outQRanges$ に新たな検索範囲 $NewQRanges$ を登録する（ステップ S 1 7 0 9）。

50

【0215】

そして、範囲分割手段A2016は、範囲内検索範囲集合inQRangesと範囲外検索範囲集合outQRangesをメッセージ転送手段2013に返す(ステップS1710)。範囲内検索範囲集合inQRangesがメッセージ転送手段2013によって範囲内検索範囲集合inRangesとして取得され、範囲外検索範囲集合outQRangesがメッセージ転送手段2013によって範囲外検索範囲集合outRangesとして取得される。

【0216】

次に、検索範囲左ビットシフト手段2015が実行する処理(ステップS1608)について説明する。図15は、この実施の形態3の検索範囲左ビットシフト手段が実行する処理を示すフローチャートである。なお、検索範囲qRangeは、仮想検索範囲iRangeと論理識別子範囲kRangeとシフト回数cとから構成されている。

10

【0217】

まず、検索範囲左ビットシフト手段2015は、シフト回数cを増やす(ステップS2101)。次いで、検索範囲左ビットシフト手段2015は、仮想範囲iRangeの上限値iMax(最大仮想論理識別子)に、それを左ビットシフトし下位ビットに、論理識別子範囲kRangeの上限値kMax(最大論理識別子)の先頭からcビット目を加えたビット列を与える(ステップS2102)。次に、検索範囲左ビットシフト手段2015は、仮想範囲iRangeの下限値iMin(最小仮想論理識別子)に、それを左ビットシフトし下位ビットに、論理識別子範囲kRangeの下限値kMin(最小論理識別子)の先頭からcビット目を加えたビット列を与える(ステップS2103)。そして、処理後の値をメッセージ転送手段2013に返す。

20

【0218】

次に、この実施の形態3におけるデータ検索システムの動作の具体例(データ検索処理の手順の具体例)について、図16を用いて説明する。

【0219】

図16は、この実施の形態3におけるピア間でのメッセージの流れの具体例を示す説明図である。

【0220】

図16において、オーバーレイネットワーク上に参加している複数のピアを示す。論理識別子13であるピア1916が、検索式Queryとして{7 温度 11}を実行するものとし、温度という属性は論理識別子範囲と対応するものとする。この例は、図9に示した例と同様である。

30

【0221】

この場合、ピア1916の登録・検索実行手段2019は、検索式Queryから対応する論理識別子範囲idRangesを算出し、論理識別子範囲[7, 11]が得られる。次に、登録・検索実行手段2019は、結果リストResultsを初期化して、他ピアからの結果を受け付けるプロセスを立ち上げ、タイマにタイムアウトを設定する(ステップS1001~S1003参照)。その後、検索範囲集合qRanges(図16の1920)を初期化する。

40

【0222】

検索範囲集合の初期化は、検索式から導かれる論理識別子範囲集合idRanges内の論理識別子範囲idRangeを、検索範囲の論理識別子範囲(図16の1920の右欄)に設定し、シフト回数c(図16の1920の左欄)を0とすれば、仮想論理識別子範囲(図16の1920の中欄)は任意の値でもよいが、好適には、仮想論理識別子範囲は、自身の論理識別子(図16に示す例では13)よりも大きくSuccessorの論理識別子(図16に示す例では15)以下の範囲に属し、かつその上限値である仮想範囲上限値の下位ビットにidRangeの上限値の先頭ビットを多く持ち、下限値である仮想範囲下限値の下位ビットにidRangeの下限値の先頭ビットを多く持たせる。ここでは、ピア1916がシフト回数2、仮想範囲[14, 14]、論理識別子範囲[8, 1

50

1]である検索範囲1921と、シフト回数1、仮想範囲[14, 14]、論理識別子範囲[7, 7]である検索範囲1922の2つの検索範囲に初期化される。

【0223】

図16において、ピア1916は、ピア1916の論理識別子15より大きくサクセサ(ピアs)の論理識別子15以下の範囲(13, 15]を用いて、検索範囲1921と検索範囲1922をそれぞれ論理識別子範囲(検索範囲1921, 1922の右欄)をもとに範囲分割する(ステップS1603)。それぞれその論理識別子範囲は(13, 15]に含まれないため、未発見検索範囲集合に格納される(ステップS1604)。

【0224】

次に、ピア1916は、この未発見検索範囲集合に含まれる検索範囲1921と検索範囲1922のそれぞれについて、ピア1916の論理識別子15より大きくサクセサ(ピアs)の論理識別子15以下の範囲(13, 15]を用いて、仮想範囲を元に範囲分割を行う(ステップS1606)。それぞれの仮想範囲はいずれも[14, 14]であり、この範囲に含まれるため、範囲内検索範囲集合に格納される。ピア1916は、この集合内の検索範囲に対して、左ビットシフトを行うことで(ステップS1608)、図16に示す検索範囲1923、検索範囲1924を得る。そして、これらはdノード転送検索範囲集合に格納される(ステップS1609)。発見検索範囲集合とsノード転送検索範囲集合の要素数は0であるが(ステップS1610のY、ステップS1614のY)、dノード検索範囲集合には、検索範囲1923、検索範囲1924が格納されているため(ステップS1612のN)、ピア1916は、これをdノードであるピア1915に転送し、ピア1915で検索範囲1923、検索範囲1924を含む検索範囲集合を探させる(ステップS1613)。

【0225】

次に、ピア1915は、検索範囲1923、検索範囲1924を含む検索範囲集合を探すピア1915は、ピア1915の論理識別子10より大きくサクセサ(ピアs)の論理識別子13以下の範囲(10, 13]で、論理識別子範囲をもとに範囲分割を行う(ステップS1603)。検索範囲1923の論理識別子範囲は[8, 11]であるため、範囲内検索範囲集合としてその論理識別子範囲に[11, 11]を含む検索範囲1925が格納され、範囲外検索範囲集合としてその論理識別子範囲に[8, 10]を含む検索範囲1926と論理識別子範囲が[7, 7]である検索範囲1924が格納される。検索範囲1925は発見検索範囲集合に格納され、検索範囲1926および検索範囲1924は未発見検索範囲集合に格納される(ステップS1604)。

【0226】

そして、ピア1915は、この未発見検索範囲集合に属する検索範囲1926および検索範囲1924のそれぞれは、ピア1915の論理識別子10より大きくサクセサ(ピアs)の論理識別子13以下の範囲(10, 13]を用いて、仮想範囲をもとに範囲分割される(ステップS1606)。それぞれの仮想範囲が[12, 13]と[13, 13]であることから、これらは範囲内検索範囲集合に格納され、左ビットシフトされて、検索範囲1926、1927となる(ステップS1608)。ピア1915は、そのサクセサ(sノード)であるピア1916に対して、発見検索範囲集合に格納された検索範囲1925をローカル検索するように指示する(ステップS1611)。また、dノードであるピア1912に対して、検索範囲1926と検索範囲1927を転送する(ステップS1613)。

【0227】

次に、検索範囲1926と検索範囲1927を受け付けたピア1912は、ピア1912の論理識別子3より大きくサクセサ(ピアs)の論理識別子6以下の範囲(3, 6]で、論理識別子範囲をもとに範囲分割を行う(ステップS1603)。検索範囲1926と検索範囲1927の論理識別子範囲はそれぞれ[8, 11]と[11, 11]であるため、いずれも範囲外検索範囲集合に格納される。また、同範囲で仮想範囲に基づく範囲分割が実行されるが(ステップS1606)、ここでもその仮想範囲が[8, 10]と[7,

10

20

30

40

50

7]であるため、sノード転送検索範囲集合に格納され(ステップS1609)、sノードであるピア1913に転送される(ステップS1615)。

【0228】

次に、検索範囲1928と検索範囲1927を受け付けたピア1913は、ピア1913の論理識別子6より大きくサクセサ(ピアs)の論理識別子9以下の範囲(6, 9]で、論理識別子範囲をもとに範囲分割を行う(ステップS1603)。検索範囲1928の論理識別子範囲は[8, 10]であるため、その論理識別子範囲が[10, 10]である検索範囲1930と、その論理識別子範囲が[8, 9]である検索範囲1931に分割され、検索範囲1930は範囲外検索範囲集合に格納され、検索範囲1931は範囲内検索範囲集合に格納される。また、検索範囲1929の論理識別子範囲は[11, 11]であるため、範囲内検索範囲集合に格納される。検索範囲1931と検索範囲1929は発見検索範囲集合に格納されるため(ステップS1604)、これらの範囲がピア1913のサクセサ(ピアs)であるピア1914でローカル検索される(ステップS1611)。また、検索範囲1931については、サクセサ(ピアs)であるピア1914に検索指示される(ステップS1615)。

10

【0229】

次に、検索範囲1930を受け付けたピア1914は、ピア1914の論理識別子9より大きくサクセサ(ピアs)の論理識別子10以下の範囲(9, 10]で、論理識別子範囲をもとに範囲分割を行う(ステップS1603)。検索範囲1930は(9, 10]に含まれるため、発見検索範囲集合に格納され(ステップS1604)、サクセサであるピア1915でローカル検索される(ステップS1611)。

20

【0230】

以上に説明したように、この実施の形態3では、検索式に合致する集合を論理識別子空間における範囲に変換し、この論理識別子範囲に属する論理識別子を担うピアに、検索メッセージが行き渡るように、論理識別子範囲とともに検索メッセージを分割する。このとき、検索対象の論理識別子範囲に対する左ビットシフトの実行、あるいはシフト回数の増加を行い、それと同時に仮想範囲の上限値および下限値を左ビットシフトさせつつ下位ビットに論理識別子範囲の上限値および下限値のシフト回数ビット目のビットを割り当てる。このような手法を用いることで、de Bruijnグラフ上での範囲検索と同様に、定数次数と対数ホップ数を実現しつつオーバーレイネットワーク上での範囲検索を実現することができる。さらに、複数の仮想ID(仮想ノード)を集合(仮想論理識別子範囲)として管理することにより、範囲検索を実行する際のオーバーレイネットワーク上の各ピアが扱うデータ量を削減することができる。

30

【0231】

上記の実施の形態2, 3において、ピアは、例えばサーバやパーソナルコンピュータなどの計算機(端末)で実現されている。また、各ピアにおいて、CPUなどの制御部がハードディスクなどの記憶手段に記憶されているプログラム(データ検索用プログラム)に従って実施の形態2, 3で説明した処理を実行することにより各構成部(手段)が実現される。また、各ピアにおいて、格納部、テーブルなどはハードディスクなどの記憶手段で実現される。

40

【0232】

また、上記の実施の形態2, 3において、各ピア(各ノード)にデータ(例えば図6、図11に示すローカルデータ格納部811に格納されるデータ)を格納する処理については、Koordeのアルゴリズムに従って行われるものとする。

【産業上の利用可能性】

【0233】

本発明は、広域でデータを共有するリレーショナルデータベースに適用できる。また、特定の属性データが発生した際に通知を受けることを可能とする広域メッセージ通知サービスにも適用できる。特に、Radio Frequency Identifier(RFID)データの格納といった用途や、システム運用管理におけるログの格納といった

50

用途や、メールやWebページの格納、Social Network Service (SNS) データの格納などにも適用可能である。また、メッセージ通知システムとしては、株価通知システムや、温度などのRFIDデータ値の変化を検知するといった用途にも利用可能である。

【図面の簡単な説明】

【0234】

【図1】この実施の形態1におけるデータ検索システムの構成を示すブロック図である。

【図2】この実施の形態1のデータ検索手順を示すフローチャートである。

【図3】左ビットシフト範囲分割処理の手順を示すフローチャートである。

【図4】この実施の形態1におけるde Bruijnグラフを示す説明図である。

10

【図5】この実施の形態1における左ビットシフト範囲分割処理のアルゴリズムを視覚的に示す説明図である。

【図6】この実施の形態2におけるデータ検索システムの構成を示すブロック図である。

【図7】この実施の形態2のデータ検索システムの全体動作を示すフローチャートである。

。

【図8】この実施の形態2のデータ検索手順を示すフローチャートである。

【図9】この実施の形態2におけるピア間でのメッセージの流れの具体例を示す説明図である。

【図10】この実施の形態2における左ビットシフト範囲分割処理のアルゴリズムを視覚的に示す説明図である。

20

【図11】この実施の形態3におけるデータ検索システムの構成を示すブロック図である。

。

【図12】この実施の形態3のデータ検索手順を示すフローチャートである。

【図13】この実施の形態3の範囲分割手段Aが実行する処理を示すフローチャートである。

【図14】この実施の形態3の範囲分割手段Bが実行する処理を示すフローチャートである。

【図15】この実施の形態3の検索範囲左ビットシフト手段が実行する処理を示すフローチャートである。

【図16】この実施の形態3におけるピア間でのメッセージの流れの具体例を示す説明図である。

30

【図17】第1の従来技術のデータ検索手順を示すフローチャートである。

【図18】de Bruijnグラフを示す説明図である。

【図19】第2の従来技術のシステム構成を示すブロック図である。

【図20】第2の従来技術のデータ検索手順を示すフローチャートである。

【図21】第2の従来技術におけるピア間でのメッセージの流れの具体例を示す説明図である。

【符号の説明】

【0235】

110, 120, 130 ノード

40

115 CPU

810, 820, 830 ピア

811 ローカルデータ格納部

812 ルーティングテーブル

813 メッセージ転送手段

814 通信手段

815 範囲分割手段

816 左ビットシフト範囲分割手段

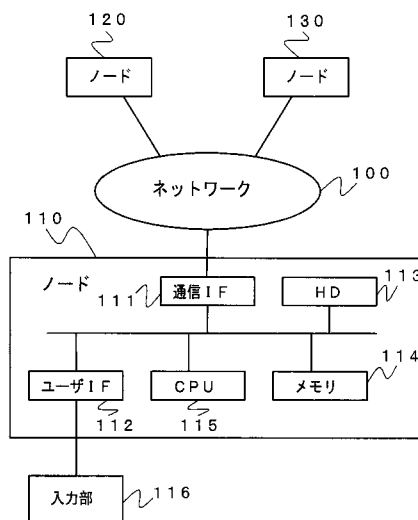
817 登録・検索実行手段

2010, 2020, 2030 ピア

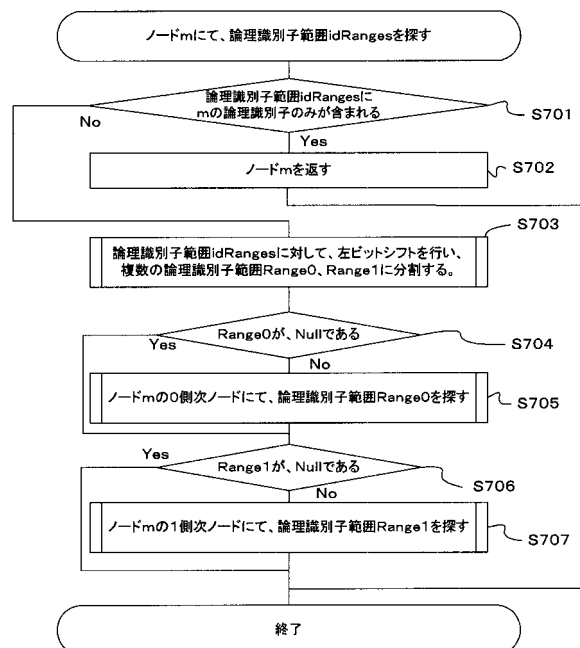
50

- 2 0 1 1 ローカルデータ格納部
- 2 0 1 2 ルーティングテーブル
- 2 0 1 3 メッセージ転送手段
- 2 0 1 4 通信手段
- 2 0 1 5 検索範囲左ビットシフト手段
- 2 0 1 6 範囲分割手段 A
- 2 0 1 7 範囲分割手段 B
- 2 0 1 8 ピア論理識別子格納部
- 2 0 1 9 登録・検索実行手段

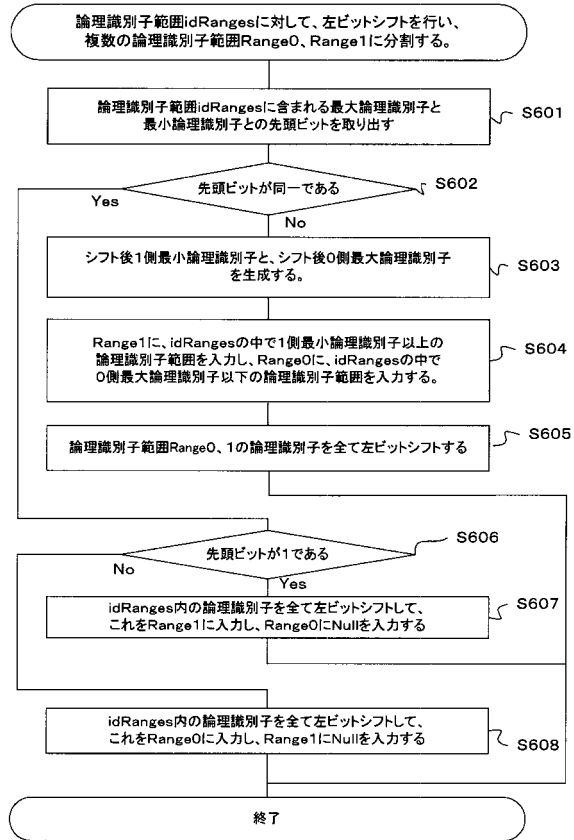
【図 1】



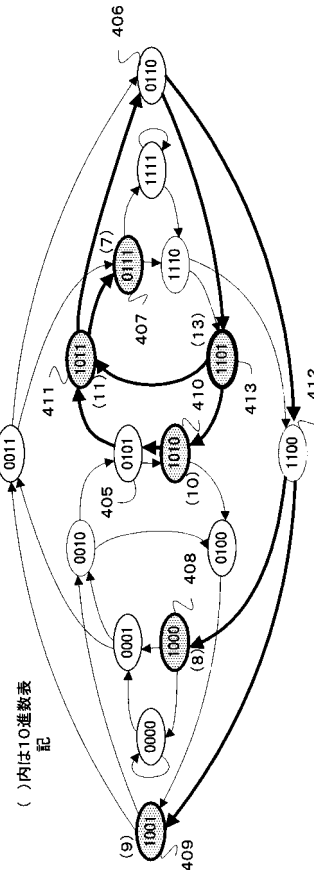
【図 2】



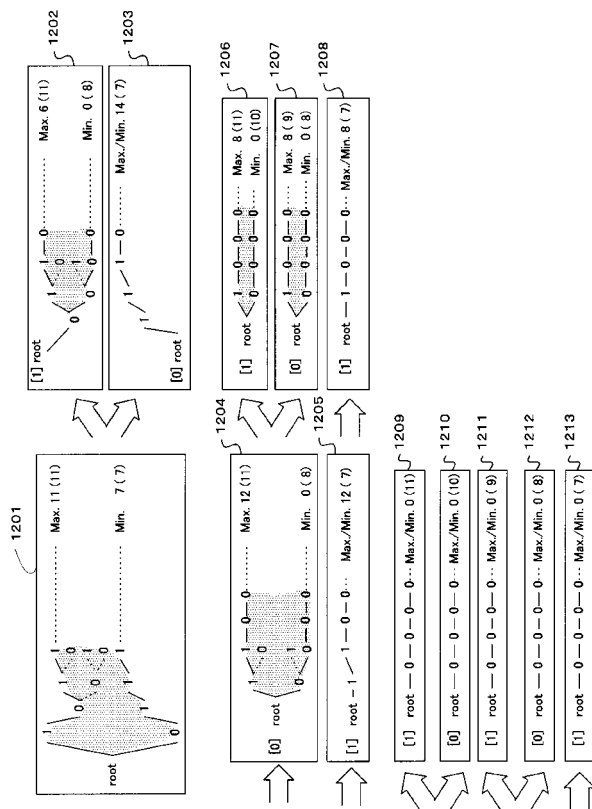
【図 3】



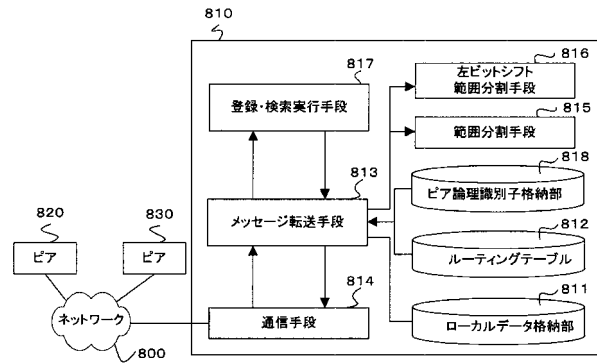
【図 4】



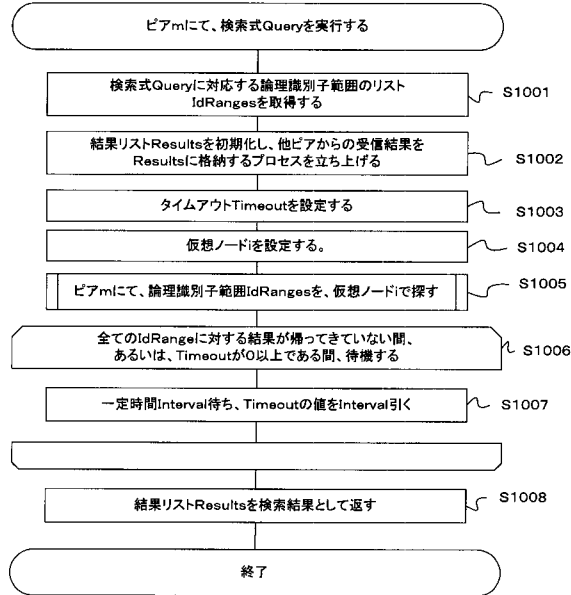
【図 5】



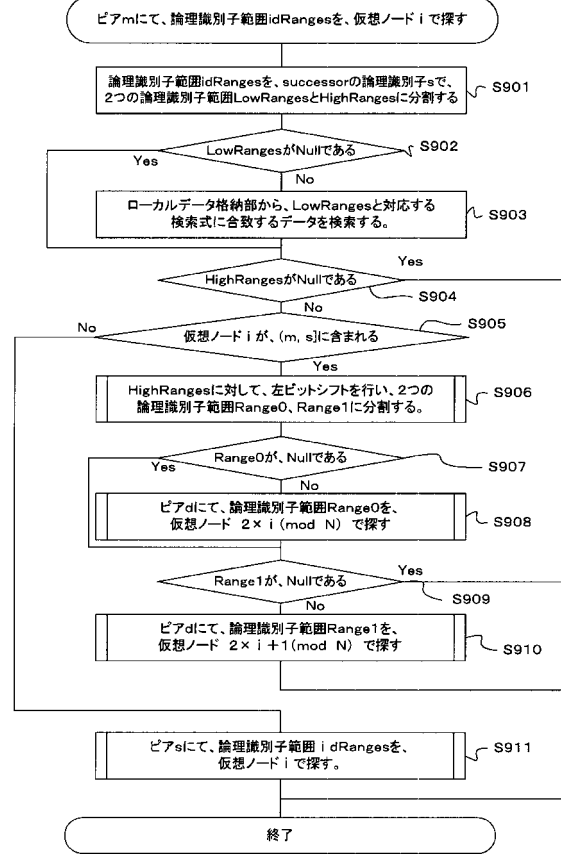
【図 6】



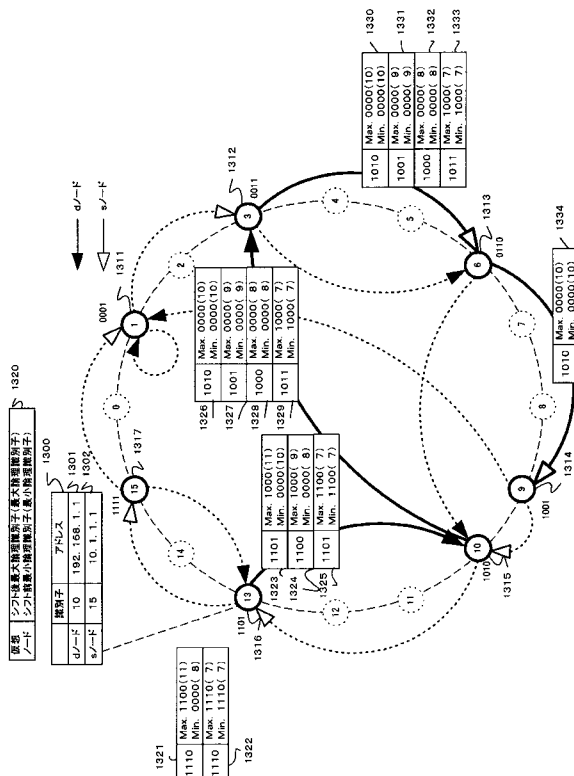
【図 7】



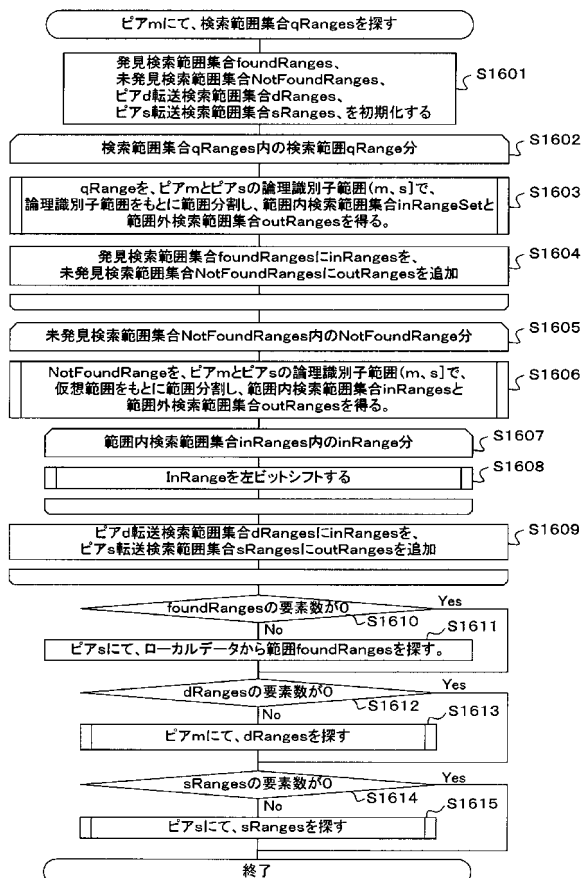
【図 8】



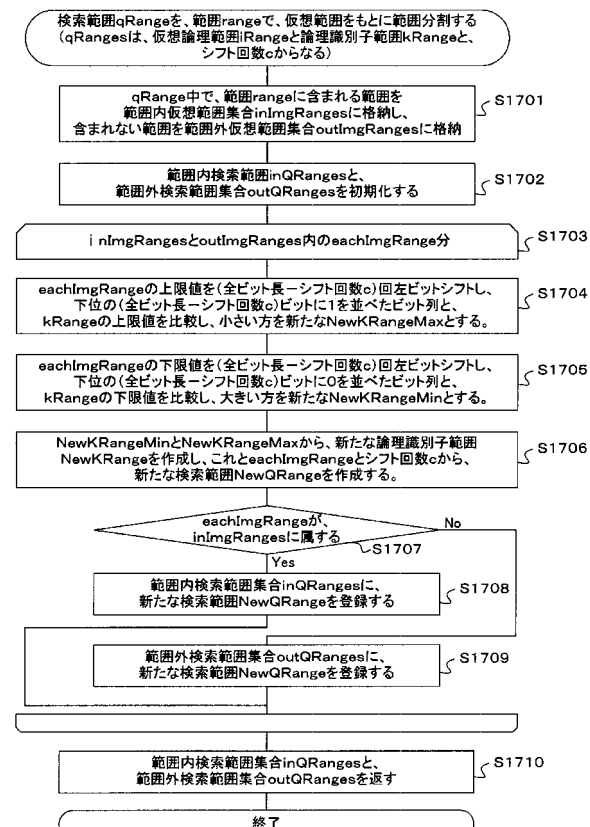
【図 9】



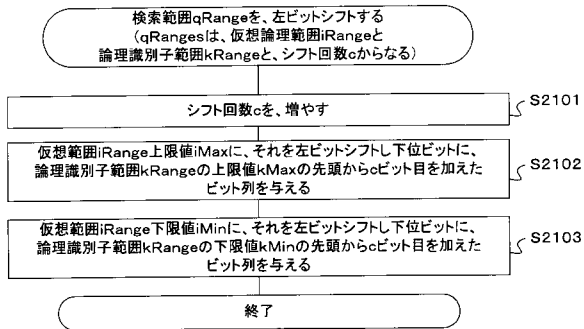
【 ㊤ 1 2 】



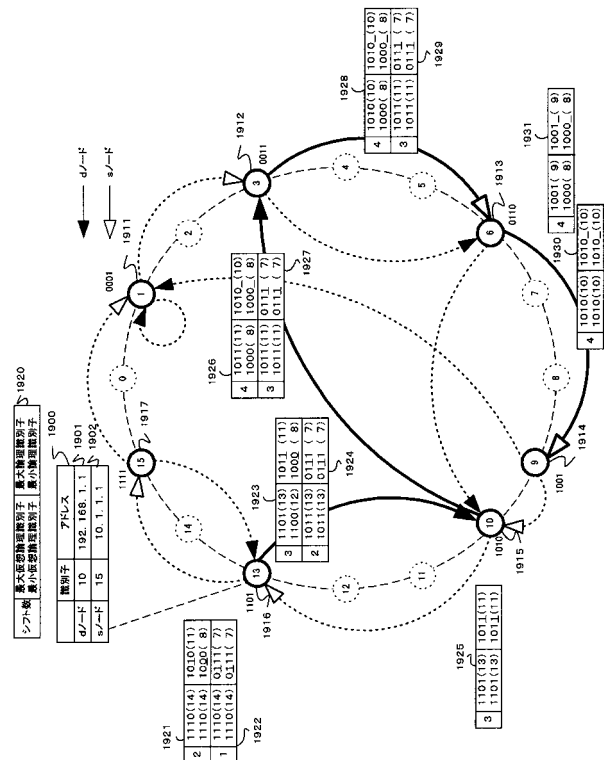
【 ㄣ 1 4 】



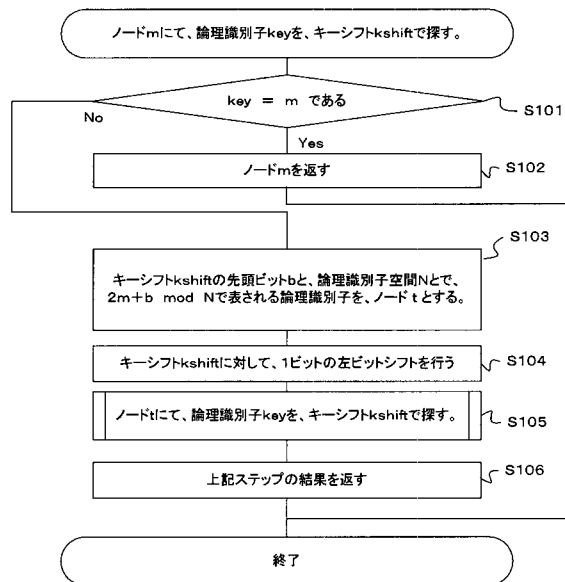
【 図 1 5 】



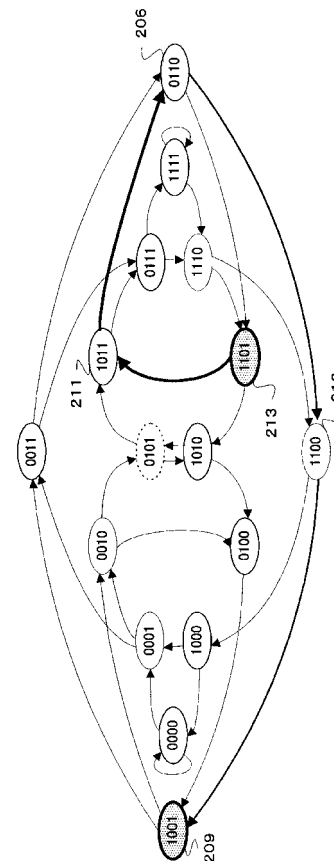
【 図 1 6 】



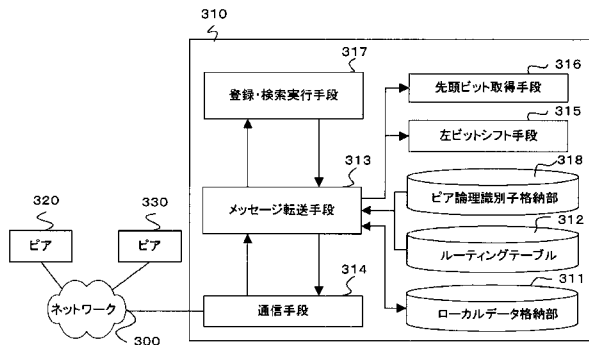
【 図 1 7 】



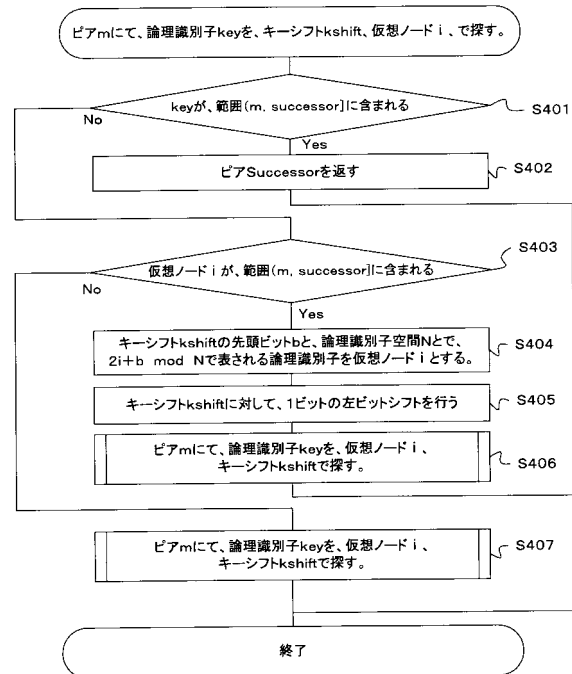
【 図 1 8 】



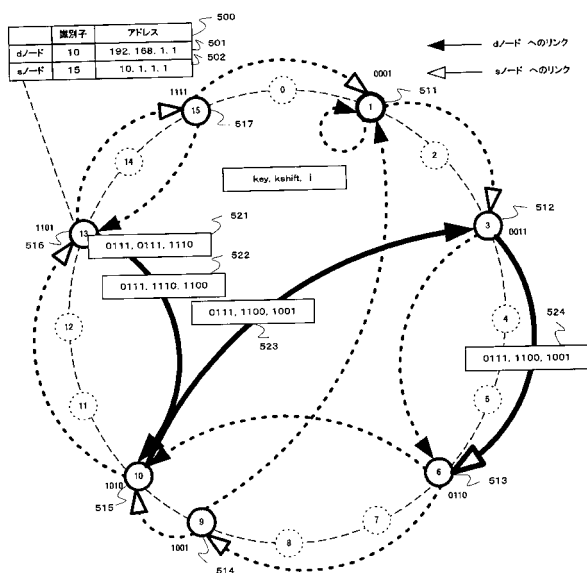
【図 19】



【図 20】



【図 21】



フロントページの続き

(56)参考文献 特開2008-269141(JP,A)

特開2008-234563(JP,A)

Anwitaman Datta, Manfred Hauswirth, Renault John, Roman Schmidt, Karl Aberer, Range Queries in Trie-Structured Overlays, Fifth IEEE International Conference on Peer-to-Peer Computing, 米国, IEEE, 2005年, pp57-66

M. Frans Kaashoek, David R. Karger, Koorde: A simple degree-optimal distributed hash table, Peer-to-Peer Systems II, 2003年10月7日, pp98-107

中台慎二 Shinji Nakadai, 谷口 邦弘 Kunihiro Taniguchi, PLATON: 超分散環境におけるデータ共有のためのP2P多次元範囲検索システム PLATON: Multi-dimensional, マルチメディア、分散、協調とモバイル(DICOMO2007), 日本, 社団法人情報処理学会, 2007年7月4日, pp173-184

Min Cai, Martin Frank, Jinbo Chen, Pedro Szekely, MAAN: A Multi-Attribute Addressable Network for Grid Information Services, Journal of Grid Computing, 米国, Kluwer Academic Publishers, 2005年2月22日, Volume 2, Number 1, pp3-14

Artur Andrzejak, Zhichen Xu, Scalable, Efficient Range Queries for Grid Information Services, Peer-to-Peer Computing 2002, 米国, IEEE, 2002年12月10日, pp30-40

Changxi Zheng, Guobin Shen, Shipeng Li, Scott Shenker, Distributed Segment Tree: Support of Range Query and Cover Query over DHT, IPTPS 2006, 米国, 2006年2月27日

(58)調査した分野(Int.Cl., DB名)

G06F 13/00

G06F 17/30