

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5965353号  
(P5965353)

(45) 発行日 平成28年8月3日(2016.8.3)

(24) 登録日 平成28年7月8日(2016.7.8)

(51) Int.Cl. F 1  
**G 0 6 F 17/30 (2006.01)**  
 G 0 6 F 17/30 4 1 4 A  
 G 0 6 F 17/30 3 3 0 A

請求項の数 7 (全 18 頁)

(21) 出願番号	特願2013-103737 (P2013-103737)	(73) 特許権者	000004226
(22) 出願日	平成25年5月16日 (2013.5.16)		日本電信電話株式会社
(65) 公開番号	特開2014-225113 (P2014-225113A)		東京都千代田区大手町一丁目5番1号
(43) 公開日	平成26年12月4日 (2014.12.4)	(74) 代理人	100107766
審査請求日	平成27年7月28日 (2015.7.28)		弁理士 伊東 忠重
		(74) 代理人	100070150
			弁理士 伊東 忠彦
		(74) 代理人	100124844
			弁理士 石原 隆治
		(72) 発明者	川原 亮一
			東京都千代田区大手町二丁目3番1号 日
			本電信電話株式会社内
		(72) 発明者	斎藤 洋
			東京都千代田区大手町二丁目3番1号 日
			本電信電話株式会社内

最終頁に続く

(54) 【発明の名称】 アドレス解決システム及び方法

(57) 【特許請求の範囲】

【請求項1】

情報通信ネットワークに接続されているobjectが、複数の属性に対する値を有し、該属性の値の組み合わせにより条件指定を行うことでobjectを検索し、該objectのアドレスを特定するアドレス解決システムであって、

リゾルバとobject管理サーバを有し、

前記リゾルバは、

各objectのアドレス及び多属性情報を含む登録要求を取得し、該登録要求に含まれる属性の階層の値を用いて、属性 $i=i_1, i_2, \dots, i_k, \dots, i_K$ とし、属性 $i_k$ に対して階層 $m_k$ 番目まで値を有するものとし、属性 $i$ の階層 $m$ における値 $x_{i,m}$ を用いて、objectの名前となる

【数9】

$$V = [x_{i_1,1} / x_{i_1,2} / \dots / x_{i_1,m_1}, x_{i_2,1} / x_{i_2,2} / \dots / x_{i_2,m_2}, \dots, x_{i_K,1} / x_{i_K,2} / \dots / x_{i_K,m_K}]$$

という値の組から構成されるobjectベクトルを生成し、該objectのアドレスと共に前記object管理サーバに送出する名前付与手段と、

取得した検索要求に含まれる各属性に関する条件が指定された属性の階層の値を用いてクエリベクトルを生成して、前記object管理サーバに送出するクエリベクトル生成手段と、  
、  
を有し、

前記object管理サーバは、

各属性と該属性の値の階層に対し、該属性と該属性の値の階層を組み合わせ、階層毎にobjectを管理するためのノードを有するツリーである組み合わせ管理ツリーを格納したobjectデータベースと、

前記リゾルバから前記objectベクトルを取得し、該objectベクトルに基づいて該組み合わせ管理ツリーを参照して、該組み合わせ管理ツリーのノードに対応付けてobjectのアドレスを登録するobject登録手段と、

10

前記リゾルバから前記クエリベクトルを取得し、該クエリベクトルに基づいて、前記組み合わせ管理ツリーを参照して、該組み合わせ管理ツリーのノードに対応するobjectのアドレスを取得するobject検索手段と、

を有することを特徴とするアドレス解決システム。

【請求項2】

前記object登録手段は、

前記組み合わせ管理ツリーに沿って、前記名前ベクトルで指定された値を担当する全てのノードにobjectのアドレスを登録する手段、

または、

20

前記組み合わせ管理ツリーのrootから、

【数10】

$$x_{i_1,1} / x_{i_2,2} / \dots / x_{i_1,m_1} / x_{i_2,1} / x_{i_2,2} / \dots / x_{i_2,m_2} / \dots / x_{i_K,1} / x_{i_K,2} / \dots / x_{i_K,m_K}$$

の順に辿っていき、到達した最後のノードのみに前記objectのアドレスを登録する手段、  
のいずれかを含む請求項1記載のアドレス解決システム。

30

【請求項3】

前記object検索手段は、

前記組み合わせ管理ツリーにおいて、rootから

【数11】

$$x_{i_1,1} / x_{i_2,2} / \dots / x_{i_1,m_1} / x_{i_2,1} / x_{i_2,2} / \dots / x_{i_2,m_2} / \dots / x_{i_K,1} / x_{i_K,2} / \dots / x_{i_K,m_K}$$

40

の順に辿っていき、最後に到達したノードで管理しているobjectを読み出す手段、

または、

前記組み合わせ管理ツリーに沿って、前記クエリベクトルで指定された値を担当するノードを全て訪問し、それらのノードを頂点とした部分木に属するノードで管理しているobjectを読み出す手段、

のいずれかを含む請求項1記載のアドレス解決システム。

【請求項4】

前記object検索手段は、

50

前記組み合わせ管理ツリーの前記ノードが、該組み合わせ管理ツリーを構築する際に用いた属性（1次属性）以外の属性（2次属性）の情報も保持している場合に、

前記リゾルバから取得した前記1次属性に関するクエリベクトルに基づいて、前記組み合わせ管理ツリーのrootから

【数12】

$$x_{i_1,1} / x_{i_2,2} / \dots / x_{i_1,m_1} / x_{i_2,1} / x_{i_2,2} / \dots / x_{i_2,m_2} / \dots / x_{i_K,1} / x_{i_K,2} / \dots / x_{i_K,m_K}$$

10

の順に進んでいき、最後に到達したノードで管理しているobjectを読み出し、該objectが前記リゾルバから取得した前記2次属性に関する条件を満たすかチェックする手段、

または、

前記リゾルバから取得した前記1次属性に関するクエリベクトルに基づいて、前記組み合わせ管理ツリーに沿って、前記クエリベクトルで指定された値を担当するノードを全て訪問し、それらのノードを頂点とした部分木に属するノードで管理しているobjectを読み出し、該objectが前記リゾルバから取得した前記2次属性に関する条件を満たすかチェックする手段、

20

のいずれかを含む請求項1記載のアドレス解決システム。

【請求項5】

前記組み合わせ管理ツリーを、

属性 $i$  ( $i=1$ から $d$ )の値の階層構造として、1階層目の値として $L_{i,1}$ 通り、1から2階層目までの値として $L_{i,2}$ 通り、1から $M_i$ 階層目の値として $L_{i,M_i}$ 通りの値を取りうるものとし、該属性 $i$ の階層1から $m$ までにおいて取りうる値の集合を $U_{i,m}$ とし、

1階層目には、属性 $i$ に対する1階層目の各値 $x_{i,1}$   $U_{i,1}$ を担当するノードを用意し、

2階層目においては、前記1階層目のノード $x_{i,1}$ を親とし、属性 $i$ の2階層目までの値が $x_{i,1}/x_{i,2}$   $U_{i,2}$ となる子ノード、ならびに属性 $j$  ( $j>i$ )の1階層目の値が $x_{j,1}$   $U_{j,1}$ となる子ノードを生成し、

30

3階層目には、前記2階層目のノードがノード $x_{i,1}/x_{i,2}$ の配下にノード $x_{i,1}/x_{i,2}/x_{i,3}$ および属性 $j$  ( $j>i$ )の1階層目の値が $x_{j,1}$ であるノードが子ノードとして生成され、かつ該2階層目のノードがノード $x_{i,1}/x_{j,1}$ の配下には属性 $j$ の2階層目までの値が $x_{j,1}/x_{j,2}$ となるノード、および属性 $k$  ( $k>j$ )の1階層目の値が $x_{k,1}$ となる子ノードを生成する処理を繰り返すことにより生成する第1の組み合わせ管理ツリー生成手段を更に有する

請求項1記載のアドレス解決システム。

【請求項6】

前記組み合わせ管理ツリーを、

属性間で優先順位付けを行い、優先度が高い順に属性を並び替え、

40

1階層目には、属性1の1階層目の値のみを列挙し、

2階層目には、属性1の2階層目の値、または属性2の1階層目の値を列挙し、属性3以降の1階層目の値は含めず、

3階層目には、属性1の3階層目の値、または属性2の1階層目または2階層目の値、または属性3の1階層目の値を列挙し、属性4以降の1階層目の値は含めない、処理を繰り返し実施して生成する第2の組み合わせ管理ツリー生成手段を更に有する

請求項1記載のアドレス解決システム。

【請求項7】

情報通信ネットワークに接続されているobjectが、複数の属性に対する値を有し、該属性の値の組み合わせにより条件指定を行うことでobjectを検索し、該objectのアドレスを

50

特定するアドレス解決方法であって、

リゾルバとobject管理サーバを有するシステムにおいて、

前記object管理サーバが

各属性と該属性の値の階層に対して該属性と該属性の値の階層を組み合わせ、階層毎にobjectを管理するためのノードを有するツリーである組み合わせ管理ツリーを格納したobjectデータベースを有し、

前記リゾルバにおいて、

各objectのアドレス及び多属性情報を含む登録要求を取得し、該登録要求に含まれる属性の階層の値を用いて、属性 $i=i_1, i_2, \dots, i_k, \dots, i_K$ とし、属性 $i_k$ に対して階層 $m_k$ 番目まで値を有するものとし、属性 $i$ の階層 $m$ における値 $x_{i,m}$ を用いて、objectの名前となる

【数 1 3】

10

$$V = [x_{i_1,1} / x_{i_1,2} / \dots / x_{i_1,m_1}, x_{i_2,1} / x_{i_2,2} / \dots / x_{i_2,m_2}, \dots, x_{i_K,1} / x_{i_K,2} / \dots / x_{i_K,m_K}]$$

という値の組から構成されるobjectベクトルを生成し、該objectのアドレスと共に前記object管理サーバに送出する名前付与ステップと、

20

前記object管理サーバにおいて、

前記リゾルバから前記objectベクトルを取得し、該objectベクトルに基づいて前記組み合わせ管理ツリーを参照して、該組み合わせ管理ツリーのノードに対応付けてobjectのアドレスを登録するobject登録ステップと、

を行う登録過程と、

前記リゾルバにおいて、

取得した検索要求に含まれる各属性に関する条件が指定された属性の階層の値を用いてクエリベクトルを生成して、前記object管理サーバに送出するクエリベクトル生成ステップと、

前記object管理サーバにおいて、

30

前記リゾルバから前記クエリベクトルを取得し、該クエリベクトルに基づいて、前記組み合わせ管理ツリーを参照して、該組み合わせ管理ツリーのノードに対応するobjectのアドレスを取得するobject検索ステップと、

を行う検索過程と、

からなることを特徴とするアドレス解決方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、アドレス解決システム及び方法に係り、特に、情報通信ネットワークにおいて、いくつかの条件指定によって、ネットワークに接続されているobject（通信機能もつ家電品や端末などを指す）を特定し、通信できるようにするためのアドレス解決システム及び方法に関する。

40

【背景技術】

【0002】

従来の情報通信ネットワークは、論理空間上のアドレスにより通信対象を指定している。固定電話では電話番号、インターネットではIPアドレスがそれにあたり、これらはネットワーク上の位置とも対応しているので、その位置に接続することで通信が可能となる。

【0003】

一方で、携帯電話の電話番号やインターネットでのURLは、より上位の論理アドレスまたは名前と解釈できる。つまり、これらは通信相手や情報の論理位置を示すが、ネットワ

50

ーク上の接続位置は示していないため、前者ではLR(ロケーションレジスタ)、後者ではDNS(ドメインネームサーバ)により、ネットワーク上の接続位置を示すアドレスへの変換(いわゆる名前解決)が必要となる。後者の場合であれば、URLから変換後のIPアドレスを用い、接続する(例えば、非特許文献1, 2参照)。

【先行技術文献】

【非特許文献】

【0004】

【非特許文献1】弓場英明監修、未来ねっと技術シリーズ コピキタスネットワーク技術、電気通信協会、2003.

【非特許文献2】<http://www.nic.ad.jp/ja/newsletter/No22/080.html>

10

【発明の概要】

【発明が解決しようとする課題】

【0005】

ところで、今後、非常に多数のobjectがネットワークに接続されるようになった時に、個々のobjectに一意に付与された名前をユーザが把握した上で通信を行う代わりに、例えば、ユーザが目の前で落し物をした自転車に乗っている人と通信したい、あるいは、ある地域である時間帯に温度が20度以上を示したセンサと通信したい、といったユーザの通信要求条件(様々な属性に対する検索条件の組み合わせ)を直接指定することでobjectを特定して通信できるようにしたい場合が想定される。

【0006】

20

しかし、既存のシステムでは、個々のobjectに一意に付与された名前をユーザが把握した上でその名前を指定して通信を行う。例えば、携帯電話であれば電話番号をユーザが指定することで通信を行う。そのため、上述のような利用形態を実現することは困難である。

【0007】

また、object毎に属性(値の情報とIPアドレス)を管理しておき、属性(値の組み合わせ)で指定された条件を満たすobjectを一つずつ調べていき、条件を満たすobjectを特定してそのIPアドレスを特定するという方法も考えられるが、全objectの探索が必要となり、object数や属性数が膨大になってしまうという問題がある。

【0008】

30

本発明は、上記の点に鑑みなされたもので、objectが有する様々な属性(場所や端末種別など)に対する条件指定を行うことで、全object探索を行うことなく、条件にマッチするobjectを特定し、該objectのIPアドレスを特定可能なアドレス解決システム及び方法を提供することを目的とする。

【課題を解決するための手段】

【0009】

上記の課題を解決するため、一態様によれば、情報通信ネットワークに接続されているobjectが、複数の属性に対する値を有し、該属性の値の組み合わせにより条件指定を行うことでobjectを検索し、該objectのアドレスを特定するアドレス解決システムであって、

リゾルバとobject管理サーバを有し、

40

前記リゾルバは、

各objectのアドレス及び多属性情報を含む登録要求を取得し、該登録要求に含まれる属性の階層の値を用いて、属性 $i=i_1, i_2, \dots, i_k, \dots, i_K$ とし、属性 $i_k$ に対して階層 $m_k$ 番目まで値を有するものとし、属性 $i$ の階層 $m$ における値 $x_{i,m}$ を用いて、objectの名前となる

【0010】

【数 1】

$$V = [x_{i_1,1} / x_{i_1,2} / \dots / x_{i_1,m_1}, x_{i_2,1} / x_{i_2,2} / \dots / x_{i_2,m_2}, \dots, x_{i_K,1} / x_{i_K,2} / \dots / x_{i_K,m_K}]$$

という値の組から構成されるobjectベクトルを生成し、該objectのアドレスと共に前記object管理サーバに送出する名前付与手段と、

取得した検索要求に含まれる各属性に関する条件が指定された属性の階層の値を用いてクエリベクトルを生成して、前記object管理サーバに送出するクエリベクトル生成手段と、  
を有し、

前記object管理サーバは、

各属性と該属性の値の階層に対し、該属性と該属性の値の階層を組み合わせ、階層毎にobjectを管理するためのノードを有するツリーである組み合わせ管理ツリーを格納したobjectデータベースと、

前記リゾルバから前記objectベクトルを取得し、前記組み合わせ管理ツリーを参照して、該objectベクトルに基づいて該組み合わせ管理ツリーのノードに対応付けてobjectのアドレスを登録するobject登録手段と、

前記リゾルバから前記クエリベクトルを取得し、該クエリベクトルに基づいて、前記組み合わせ管理ツリーを参照して、該組み合わせ管理ツリーのノードに登録されているobjectのアドレスを取得するobject検索手段と、を有するアドレス解決システムを提供される。

【発明の効果】

【0011】

上述のように、本発明によれば、objectが有する様々な属性（場所や端末種別など）に対する条件指定を行うことで、全object探索を行うことなく条件にマッチするobjectを特定し、該objectのIPアドレスを特定可能な名前解決を提供することが可能となる。

【図面の簡単な説明】

【0012】

【図1】本発明が適用されるIPネットワークの基本構成例である。

【図2】本発明のリゾルバの構成例である。

【図3】本発明のobject管理サーバの構成例である。

【図4】本発明の第1の実施の形態における組み合わせ管理ツリーの例である。

【図5】本発明の第1の実施の形態における名前付与・登録時のフローチャートである。

【図6】本発明の第1の実施の形態における検索（名前解決）時のフローチャートである。

【発明を実施するための形態】

【0013】

以下、図面と共に本発明の実施の形態を説明する。

【0014】

図1は、本発明が適用されるIPネットワークの基本構成例である。

【0015】

同図に示すシステムは、IPネットワーク1にobject10、リゾルバ20、object管理サーバ30が接続されている。なお、同図では、IPネットワーク1を例に記載しているが、IPアドレス以外の論理アドレスを用いた他のネットワークであってもよい（つまり、論理アドレスがネットワーク上の位置を指し、かつその論理アドレスを指定すれば所望の宛先までデータやパケットを転送することが可能であれば他のネットワークであってもよい）。ここで、object10はobjectを識別するID、IPアドレスに加え、多属性情報（場所、端

10

20

30

40

50

未種別、時刻等)有するものとする。

【0016】

リゾルバ20は、登録・検索要求受付部21、object名前付与・クエリベクトル生成部22を有し、object名前付与・クエリベクトル生成部22はobject管理サーバ30に接続されている。

【0017】

object管理サーバ30は、object・クエリベクトル受付部31、object登録・検索部32、objectデータベース33、ツリー構築部34を有し、object・クエリベクトル受付部31はリゾルバ20に接続されている。

【0018】

上記の構成に基づいて以下に本発明の実施の形態を説明する。

【0019】

[第1の実施の形態]

本実施の形態では、管理サーバ30へ、属性の値、ツリー構造、objectのIPアドレスの登録、及び属性に対する値を指定して検索する処理について説明する。

【0020】

情報通信ネットワーク1に接続されているobject10(例えば、通信機能を持つ家電品や端末、センサなどを指し、IPアドレスを持つとする)が、複数の属性(例えば、時刻や場所など)に対する値(時刻=2013/1/31、場所=東京/武蔵野市など)を有するとする。ここで、各属性に対する値が階層構造を持つとする。例えば、「場所」という属性に対する値は、都道府県/市町村という2階層からなるものとし、具体的な値としては、例えば、あるobjectは、「東京都/武蔵野市」、という値をもつとする。このように、属性に対する値に関して、階層構造を定義する。object10は、これらの属性のうち、いくつかに対する値を持つものとする。

【0021】

リゾルバ20の登録・検索要求受付部21がobject10から属性、属性に対する値、階層構造を含む登録要求のパケットを受信し、object名前付与・クエリベクトル生成部22が、その属性と属性に対する値の組をobjectの名前として付与する(以下、「名前付与」と記す)。

【0022】

object管理サーバ30のobjectデータベース33には、上記で定義した各属性、値の階層に対し、それらの属性、値の階層の組み合わせがツリー構造で表現され保持されている。このツリーを「組み合わせ管理ツリー」と呼ぶ。object管理サーバ30のツリー構築部34は、組み合わせ管理ツリーと属性と当該属性の値を対応付けてobjectデータベース33に登録する。当該処理を「indexing」と呼ぶ。

【0023】

検索時には、object10において、いくつかの属性に対する値を指定し、それら属性と属性の値の組を入力し、リゾルバ20の登録・検索要求受付部21において、object10で指定された値を含む検索要求パケットを受信し、object名前付与・クエリベクトル生成部22において、当該検索要求パケットの値に基づいてクエリベクトルを生成し、object管理サーバ30に送り、object管理サーバ30のobject登録・検索部32においてobjectデータベース33を検索してその結果として特定のobjectのアドレスを、リゾルバ20を介してobject10に通知する。当該処理を「名前解決」と呼ぶ。

【0024】

このように、本発明では、属性と属性の値の組とツリー構造をobjectデータベース33に管理しておき、属性と属性の値の組で指定された条件を満たすobjectをツリーに沿って探索することで、全object探索することなく、objectの特定を可能にしている。

【0025】

<indexing>

上記の組み合わせ管理ツリーの具体的な構成方法(indexing)を以下に示す。

10

20

30

40

50

## 【 0 0 2 6 】

ツリー構築部 3 4 は、予め、以下で述べる組み合わせ管理ツリーを生成し、object データベース 3 3 に保持する。

## 【 0 0 2 7 】

以下にツリー構築部 3 4 の処理について説明する。

## 【 0 0 2 8 】

属性  $i$  ( $i=1$  から  $d$ ) の値の階層構造として、1 階層目の値として  $L_{i,1}$  通り、1 から 2 階層目までの値として  $L_{i,2}$  通り、1 から  $M_i$  階層目の値として  $L_{i,M_i}$  通りの値を取りうるとする。また、属性  $i$  の階層 1 から  $m$  までにおいて取りうる値の集合を  $U_{i,m}$  とする。このとき、ツリー構築部 3 4 は、上記の組み合わせ管理ツリーを以下のように構成する。

10

## 【 0 0 2 9 】

当該組み合わせ管理ツリーの 1 階層目には、属性  $i$  に対する 1 階層目の各値  $x_{i,1} \in U_{i,1}$  を担当するノードを用意する。(以下では、例えば値  $x_{i,1}$  を担当するノードをノード  $x_{i,1}$  と呼ぶことがある。) つまり、属性  $i$  に対して、1 階層目で取りうる値のパターン数  $L_{i,1}$  個のノードを用意する。属性  $i=1 \sim d$  全体で、

## 【 0 0 3 0 】

## 【 数 2 】

$$\sum_{i=1}^d L_{i,1}$$

20

個のノードとなる。

## 【 0 0 3 1 】

次に、ツリーの 2 階層目においては、ツリー 1 階層目のノード  $x_{i,1}$  (例えば属性  $i$  が場所名なら、 $x_{i,1}$ =東京) を親とし、属性  $i$  の 2 階層目までの値が  $x_{i,1}/x_{i,2} \in U_{i,2}$  (例えば、 $x_{i,1}/x_{i,2}$ =東京/武蔵野市) となる子ノード、ならびに属性  $j$  ( $j>i$ ) の 1 階層目の値が  $x_{j,1} \in U_{j,1}$  となる子ノードを生成する。つまり、ノード  $x_{i,1}$  の配下に、同じ属性  $i$  の子ノード以外に、

30

## 【 0 0 3 2 】

## 【 数 3 】

$$\sum_{j>i}^d L_{j,1}$$

40

個の子ノードが生成される。ツリーの 3 階層目には、2 階層目のノードがノード  $x_{i,1}/x_{i,2}$  の配下にノード  $x_{i,1}/x_{i,2}/x_{i,3}$  および属性  $j$  ( $j>i$ ) の 1 階層目の値が  $x_{j,1}$  であるノードが子ノードとして生成され、かつ 2 階層目のノードがノード  $x_{i,1}/x_{j,1}$  の配下には属性  $j$  の 2 階層目までの値が  $x_{j,1}/x_{j,2}$  となるノード、および属性  $k$  ( $k>j$ ) の 1 階層目の値が  $x_{k,1}$  となる子ノードが生成される。

## 【 0 0 3 3 】

以上を繰り返し実施することで組み合わせ管理ツリーを構成し、各ノードは自身が担当する値を有する object を管理する。

50

## 【 0 0 3 4 】

なお、上記の組み合わせ管理ツリーノードは、1つのノードあるいは幾つかのノードをまとめたものが一つの物理サーバ上に存在して、そのようなサーバが地理的に分散配置されていてもよい。

## 【 0 0 3 5 】

具体例を用いて上記のindexingの手順を詳しく説明する。

## 【 0 0 3 6 】

例として、属性1=場所名、属性2=端末種別名、の2つを考える。また、場所名については、都道府県名/市町村名のように2階層から成るとする。端末種別名も、大分類名/小分類名の2階層から成るとする。大分類名の値の例としては、OA機器、家電、センサ、などと  
10  
とする。小分類名の値の例としては、大分類名=OAの配下に、プリンタ、サーバ、などが存在し、家電の配下に、冷蔵庫、TV、エアコンなどが存在するとする。

## 【 0 0 3 7 】

本発明の手順に従って組み合わせ管理ツリーを構成すると図4のようになる。本図の構成について説明する。

## ・ 1階層目：

(1)左側の点線枠内には、場所名1階層目(つまり都道府県名)の値(つまり、北海道、沖縄など)が $L_{1,1}$ (=47個)個列挙されている。

## 【 0 0 3 8 】

(2)右側の点線枠内には、端末種別名1階層目(つまり大分類名)の値(つまり、OA機  
20  
器、家電など)が $L_{2,1}$ 個列挙されている。

## 【 0 0 3 9 】

## ・ 2階層目：

(1)左側の点線枠内には、場所名1階層目の値を持つ親ノードの配下に、場所名2階層目(つまり市町村名)の値(札幌市など)が列挙される。従って、本枠内には、 $L_{1,2}$ 個の都道府県・市町村名の組が列挙される。

## 【 0 0 4 0 】

(2)真ん中の点線枠内には、端末種別名1階層目の値が列挙されている。従って、本枠内には、 $L_{1,1} * L_{2,1}$ 個の都道府県名・端末種別大分類名の組が列挙される。

## 【 0 0 4 1 】

(3)右側の点線枠内には、端末種別名2階層目(つまり小分類名)の値(つまり、プリンタ、サーバなど)が列挙されている。従って、本枠内には、 $L_{2,2}$ 個の端末種別大分類・小分類名の組が列挙される。  
30

## 【 0 0 4 2 】

## ・ 3階層目：

(1)左側の点線枠内には、端末種別名1階層目の値が列挙される。従って、本枠内には、 $L_{1,2} * L_{2,1}$ 個の都道府県名・市町村名・端末種別大分類名の組が列挙される。

## 【 0 0 4 3 】

(2)右側の点線枠内には、端末種別名2階層目の値が列挙される。従って、本枠内には、 $L_{1,1} * L_{2,2}$ 個の都道府県名・端末種別大分類名・端末種別小分類名の組が列挙される。  
40

## 【 0 0 4 4 】

・ 4階層目：端末種別名2階層目の値が列挙される。従って、本枠内には、 $L_{1,2} * L_{2,2}$ 個の都道府県名・市町村名・端末種別大分類名・端末種別小分類名の組が列挙される。

## 【 0 0 4 5 】

以上のように、属性と属性の値の組を、object管理サーバ30のobjectデータベース33のツリーに登録することで、属性と属性の値の階層の組み合わせによるいずれの条件指定に対しても、対応するノードが存在するため、検索時にツリーに沿って条件指定を満たすノードを特定して該ノードで管理されているobjectを読み出すことで、条件を満たすobjectの特定が可能となる。

## 【 0 0 4 6 】

次に、図4の例を用いて、objectデータベース33内の組み合わせ管理ツリーにobjectを格納する際の具体例を説明する。

【0047】

例として、object={場所=北海道/札幌, 端末種別=OA/サーバ}の場合を考える。

【0048】

object格納法方法として2通りを考える。

【0049】

手順(1)として、まずツリー1階層目左側の枠線内の「北海道」ノードと、右側の枠線内の「OA」ノードに登録する。次に、2階層目では、左側の枠線内の「札幌」ノード、中央の枠線内の「OA」ノード、右側の枠線内の「サーバ」に登録する。3階層目では、左側の枠線内の「OA」ノード、右側の枠線内の「サーバ」に登録する。4階層目では「サーバ」ノードに登録する。すなわち、計8ノードに登録されることになる。

10

【0050】

手順(2)として、北海道/札幌/OA/サーバと辿った4階層目の「サーバ」ノードのみ登録する。

【0051】

<登録処理>

以下にobject名前付与とobjectデータベース33への登録処理について説明する。

【0052】

なお、予めobjectデータベース33には、組み合わせ管理ツリーが格納されているものとして説明する。

20

【0053】

図5は、本発明の第1の実施の形態における登録時のフローチャートである。

【0054】

ステップ101) object10がネットワーク1に接続されたら、object10は、object登録要求パケットをリゾルバ20内の登録・検索要求受付部21に送信する。その際、object自身のIPアドレス(=IP#A)、objectID、多属性情報を通知する。

【0055】

ステップ102) object10からの登録要求パケットがリゾルバ20の登録・検索要求受付部21に到着したら、当該object10の各種情報を読み出し、object名前付与・クエリベクトル生成部22に通知する。object名前付与・クエリベクトル生成部22では、多属性情報から当該objectの名前ベクトルVを生成する(名前付与)。

30

【0056】

具体的な手順は以下の通りである。

【0057】

object10が値を有する属性iが*i*=*i*<sub>1</sub>, *i*<sub>2</sub>, ..., *i*<sub>*k*</sub>, ..., *i*<sub>*K*</sub>とし、属性*i*<sub>*k*</sub>に対して階層*m*<sub>*k*</sub>番目まで値を有するものとし、属性*i*の階層*m*における値*x*<sub>*i*,*m*</sub>を用いて、objectの名前となるobjectベクトル

【0058】

【数4】

40

$$V = [x_{i_1,1} / x_{i_1,2} / \dots / x_{i_1,m_1}, x_{i_2,1} / x_{i_2,2} / \dots / x_{i_2,m_2}, \dots, x_{i_K,1} / x_{i_K,2} / \dots / x_{i_K,m_K}]$$

を生成する。

【0059】

生成したベクトルVとIP#Aの組を、object管理サーバ30へ通知する。

【0060】

50

ステップ103) object管理サーバ30内のobject・クエリベクトル受付部31が、objectベクトルVとIP#Aの組を受信したら、それをobject登録・検索部32に通知する。object登録・検索部32では、以下の(1)または(2)の手順により、objectデータベース33に登録する。

(1)objectデータベース33のツリーに沿ってVで指定された値を担当する全てのノードに該objectを登録する。

(2) objectデータベース33のツリーにおいて、rootから、

【0061】

【数5】

10

$$x_{i_1,1}/x_{i_2,2}/\dots/x_{i_1,m_1}/x_{i_2,1}/x_{i_2,2}/\dots/x_{i_2,m_2}/\dots/x_{i_K,1}/x_{i_K,2}/\dots/x_{i_K,m_K}$$

の順に辿っていき、到達した最後のノードのみに該objectを登録する。

以下に、図4を用いてobject登録時の処理を具体的に示す。

【0062】

例えば、V={場所名:北海道/札幌, 端末種別名:OA/プリンタ}の登録を考える。

【0063】

(1)の手順の場合、図4のツリーの1階層目左側の枠線内の「北海道」ノード, 1階層目右側枠線内の「OA」ノード, 2階層目左側枠線内の「札幌」ノード, 2階層目中央枠線内の「OA」ノード, 2階層目右側枠線内の「プリンタ」ノード, 3階層目左側枠線内の「OA」ノード, 3階層目右側枠線内の「プリンタ」ノード, 4階層目枠線内の「プリンタ」ノード, の計8ノードに該objectを登録する。

(2)の手順の場合、4階層目枠線内の「プリンタ」ノードのみに該objectを登録する。

< 検索処理 >

以下に、object検索(名前解決)時の処理について説明する。

【0064】

図6は、本発明の第1の実施の形態における検索(名前解決)時のフローチャートである。

20

30

【0065】

ステップ201) ユーザは端末から各属性に関する条件指定をした情報(例えば{場所名=北海道, 端末種別名=OA})を検索要求パケットとしてリゾルバ20に送信する。

【0066】

ステップ202) 検索要求パケットをリゾルバ20内の登録・検索要求受付部21にて受信したら、各属性に対する条件指定を読み出し、object名前付与・クエリベクトル生成部22へ通知する。object名前付与・クエリベクトル生成部22では、以下の手順により、クエリベクトルqを生成する。

【0067】

検索時に条件指定された属性 $i_1, i_2, \dots, i_k, \dots, i_K$ について、各 $i_k$ に対して階層1から $m_k$ に対する値が指定されているとする。これら値が指定された各階層の値を用いてクエリベクトル

40

【0068】

【数6】

$$q = [x_{i_1,1} / x_{i_1,2} / \dots / x_{i_1,m_1}, x_{i_2,1} / x_{i_2,2} / \dots / x_{i_2,m_2}, \dots, x_{i_K,1} / x_{i_K,2} / \dots / x_{i_K,m_K}]$$

を生成し、object管理サーバ30へ通知する。

【0069】

ステップ203)クエリベクトルqをobject管理サーバ30内のobject・クエリベクトル受付部31にて受信したら、それをobject登録・検索部32に通知する。object登録・検索部32では、以下の(1)または(2)の手順により、objectデータベース33を検索する。

10

(1) objectデータベース33の組み合わせ管理ツリーにおいて、rootから、

【0070】

【数7】

$$x_{i_1,1} / x_{i_2,2} / \dots / x_{i_1,m_1} / x_{i_2,1} / x_{i_2,2} / \dots / x_{i_2,m_2} / \dots / x_{i_K,1} / x_{i_K,2} / \dots / x_{i_K,m_K}$$

20

と辿って行き、最後に到達したノードで管理しているobjectを読み出す。

(2) object登録・検索部32は、objectデータベース33から、ツリーに沿ってqで指定された値を担当するノードを全て訪問し、それらノードを頂点とした部分木に属する全てのノードで管理しているobjectを読み出す。

ステップ204) object登録・検索部32は、以上の(1)または(2)の手順で特定されたobjectの集合に対し、該objectのIPアドレスをリゾルバ20に通知する。

【0071】

30

なお、登録処理時に上記の(1)の方法を用いてVで指定された値を担当する全てのノードにobject登録している場合には、検索時には、(1)の方法によりobjectを検索し、また、登録処理時に上記の(2)の方法を用いて1つのノードのみにobjectを登録している場合には、検索時には(2)の方法を用いてobjectを検索することとする。

【0072】

以下に、図4を用いてobject検索時の処理を具体的に示す。

【0073】

例えば、{場所名=北海道}のみの条件指定を考える。

【0074】

(1)の手順の場合、図4のツリーの1階層目左側の枠線内の「北海道」ノードで管理されているobjectを読み出せばよい。

40

【0075】

(2)の手順の場合、該「北海道」ノードを頂点とした部分木に属する全てのノードにアクセスし、それらノードで管理されているobjectを読み出す。

【0076】

別の例として、{場所名=北海道, 端末種別名=OA}という条件指定を考える。

【0077】

(1)の手順の場合、図4のツリーの2階層目中央の枠線内の「OA」ノードで管理されているobjectを読み出せばよい。

【0078】

50

(2)の手順の場合、図4のツリーにおいて、北海道/OA (= 2階層目の中央の枠線内)、および、北海道/(市町村名指定なし)/OA/ (= 2階層目で北海道内の各市町村ノード全てを対象として、それぞれの配下の「OA」ノードを頂点とした部分木に属する全てのノード)にアクセスし、それらノードで管理されているobjectを読み出す。

【0079】

[第2の実施の形態]

本実施の形態では、object検索を行う際に、第1の実施の形態のように組み合わせ管理ツリーで該当するノードを特定したら、そのノードで管理しているobject(の集合)に対して組み合わせ管理ツリーで管理している属性(1次属性とよぶ)以外の属性(2次属性と呼ぶ)に関する条件を指定することで通信したいobjectを絞り込む。

10

【0080】

例えば、属性の値が階層構造で表現されないような属性(センサの気温や端末の大きさなど値が連続値を取る属性の場合など)に対してobjectが値を有するとし、そのような属性についても条件指定してobjectを特定したい場合(例: 気温 20度)は、まず、組み合わせ管理ツリーを用いてそこで管理される階層の組について条件指定を行ってobjectの集合を特定し、その集合の中から、さらに気温 20度を満たすobjectに絞り込む。あるいは、本発明で述べた組み合わせ管理ツリーには条件指定される可能性の高い属性のみを用いることとし、それら属性と当該属性値の組で該当objectの集合をまず抽出し、それ以外の属性に対する指定条件については、2次属性としてもよい。

【0081】

20

本実施の形態では、上記の図6のステップ203の処理が異なる。以下、ステップ203'として説明する。

【0082】

本実施の形態では、組み合わせ管理ツリーを構成する際に用いた1次属性情報のみならず2次属性情報についても、object10を管理するノードが保有しているものとする。また、検索時には1次属性ならびに2次属性に対する条件指定があったとし、その指定条件がリゾルバ20を介して、object管理サーバ30内のobject・クエリベクトル受付部31に通知されているとする。なお、クエリベクトルは1次属性に対する条件指定により構成されているとする。

ステップ203')クエリベクトルqをobject管理サーバ30内のobject・クエリベクトル受付部31にて受信したら、それをobject登録・検索部32に通知する。object登録・検索部32では、以下の(1)または(2)の手順により、objectデータベース33を検索する。

30

(1) objectデータベース33の組み合わせ管理ツリーにおいて、rootから、

【0083】

【数8】

$$x_{i_1,1}/x_{i_2,2}/\dots/x_{i_1,m_1}/x_{i_2,1}/x_{i_2,2}/\dots/x_{i_2,m_2}/\dots/x_{i_K,1}/x_{i_K,2}/\dots/x_{i_K,m_K}$$

40

と辿って行き、最後に到達したノードで管理しているobjectを読み出す。

(2) objectデータベース33の組み合わせ管理ツリーに沿ってqで指定された値を担当するノードを全て訪問し、それらノードを頂点とした部分木に属する全てのノードで管理しているobjectを読み出す。

【0084】

上記手順で特定されたobjectの集合に対して、2次属性に関する条件を指定することで特定したいobjectを絞り込む。

【0085】

50

具体例として、objectデータベースの各ノードにおいて、2次属性に関してR-tree（非特許文献3：A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," ACM SIGMOD 1984.、非特許文献4：L. Arge et al., "The Priority R-tree: a practically efficient and worst-case optimal R-tree," SIGMOD 2004.）やKD-tree（非特許文献5：[http://en.wikipedia.org/wiki/K-d\\_tree](http://en.wikipedia.org/wiki/K-d_tree)）と呼ばれるツリー型のデータ構造で各objectの値を管理しているとする。例えば2次属性として、気温という属性と、末端の大きさという属性があったとする。その際、objectは2次元空間上にマッピングされる。この2次元空間を複数のエリアに分割してそれをroot配下の各ノードがそれぞれのエリアを担当するとし、更にそのエリアを分割して自身の子ノードが担当するというツリー構造を構築する。検索時には、この例では1次元目と2次元目それぞれに対してレンジを指定し、そのレンジと重なるノードを辿って行くことで条件を満たすobjectを特定する。

10

【0086】

このように2次属性についても検索処理を行い、その結果特定されたobjectを読み出す。

【0087】

[第3の実施の形態]

第1の実施の形態では、組み合わせ管理ツリーの1階層目において、各属性*i*に対する1階層目の値を列挙して、各々に対するノードを生成していたが、本実施の形態では、そうする代わりに、以下の手順で組み合わせ管理ツリーを構成する。

【0088】

20

まず属性間で優先順位付けを行い、優先度が高い順に属性が並び替えられているとする。つまり、属性*i*は属性*j* ( $j > i$ ) より高優先とする。ツリーの1階層目には、属性1の1階層目の値のみを列挙する。ツリーの2階層目には、属性1の2階層目の値、または属性2の1階層目の値を列挙し、属性3以降の1階層目の値は含めない。ツリーの3階層目には、属性1の3階層目の値、または属性2の1階層目または2階層目の値、または属性3の1階層目の値を列挙し、属性4以降の1階層目の値は含めない。

【0089】

以上を繰り返し実施して組み合わせ管理ツリーを構成する。

【0090】

object検索時には、第1の実施の形態では任意の属性に対して条件指定有無を選択できたのに対して、ここでは属性1から*J*までは条件指定を行うこととし、検索手順は第1の実施の形態の方法と同じとする。

30

【0091】

図4の例の場合、例えば、属性1 = 場所名、属性2 = 端末種別名とし、検索時には場所名については条件指定を必須とし、端末種別名については条件指定の有無は選択可能とする。その場合、図4の1階層目の右側の枠線内のノードと2階層目の右側の枠線内のノードは不要となる。こうすることで、管理すべきノード数を削減できる。

【0092】

以上、第1～第3の実施の形態で説明したように、本発明によれば、objectが有する様々な属性（場所や端末種別など）に対する条件指定を行うことで、全object探索を行うことなく条件にマッチするobjectを特定し、該objectのIPアドレスを特定可能な名前解決を提供することが可能となる。

40

【0093】

数値例を示す。2つの属性*i* ( $i=1,2$ )があるとし、いずれも2階層の値を持つとし、1階層目では $L_{i,1}=2^4$ 通り、2階層目までは $L_{i,2}=2^{10}$ 通りの値を持つとする。これら属性の組に対して図4の組み合わせ管理ツリーを構成したとする。また、これら2つの属性以外に8個の2次属性（各属性は連続値を値として持つ）を各objectは有するとする。また、object数 $N=2^{28}$ とする。また各objectはいずれの属性に対しても一様に分布した値を持つとする。

【0094】

50

以下に検索の例として、例 1、例 2 を示す。

【 0 0 9 5 】

例 1 )

もし検索条件として属性1に対して2階層目まで、属性2に対しても2階層目まで指定があった場合には、図4のツリーを辿る探索コストのみ(属性iの各階層の値をチェックするのみ)で、 $N=2^{28}$ 個(約2.7億)のobjectの中から、 $N \times (1/2^{10})^2=256$ 個のobjectに絞り込むことが可能となる。

【 0 0 9 6 】

例 2 )

属性1の1階層目のみ条件指定を行い、2次属性に対する範囲指定でobjectをさらに絞り込みたい場合を考える。なお2次属性に対しては、各objectは各属性に対し0から1の間の一様分布に従う値を有するとする。また、2次属性に対する検索条件は、各属性に対しレンジ長0.5のレンジ指定を行うとする。この場合、属性1での検索により、objectは $N \times 1/16=2^{24}$ ( $=N'$ とおく)に絞り込まれる。次に2次属性での検索にkd-treeのようなツリー構造を用いてレンジ指定検索を行ったとする。8つの各属性に対してレンジ長0.5のレンジ指定を行った場合、object数は $N' \times (0.5)^8=2^{16}$ (約6.5万)個に絞り込まれる。この2次属性検索(レンジ指定検索)の計算量に関し、本発明の第1の実施の形態の方法で述べた(1)の方法と(2)の方法それぞれについて計算量を評価する。

【 0 0 9 7 】

(1)の方法の場合、組み合わせ管理ツリーの属性1の1階層目に位置するノードに $N'$ 個のobjectが存在する。この $N'$ 個のobjectに対してレンジ指定による検索を行ったときの計算量を以下のように評価する。

【 0 0 9 8 】

レンジ指定用のツリーとして、各ノードが $2^8$ 個の子ノードを持ち、各子ノードは親ノードが担当する8次元空間上のエリアを $1/2^8$ 個に分割した一つのサブエリアを担当するとする。このとき、検索条件で指定された範囲と、子ノードの担当範囲の一部が重なった場合、該子ノード以降をさらに探索する必要があるため、この一部重なったノード数を計算量として評価した(それを全object数 $N$ で正規化した)。このとき、(1)の方法の場合、計算量は $9.10e-05$ であった。

(2)の方法の場合、条件を満たす属性1の1階層目のあるノードを頂点として、その配下にある該ツリーのleafノード $2^{16}$ 個各々が管理するobjectに対して、レンジ指定検索を行うことになる。この場合、各leafノード1個当り、 $N'/2^{16}=256$ 個のobjectが存在する。1leafノード当りでの計算量を $2^{16}$ 倍した値が全体での計算量となり、その計算量は $3.91e-03$ であった。

【 0 0 9 9 】

以上のように、計算量を全object探索よりも小さくできていることが分かる。特に、例2のように最初の条件指定で特定されるobject数 $N'$ が大きく、2次属性での検索を行う場合には、(1)の方法がより効果的であると考えられる。

【 0 1 0 0 】

なお、上記のリゾルバ20及びobject管理サーバ30の各構成要素の動作をプログラムとして構築し、リゾルバ、object管理サーバとして利用されるコンピュータにインストールして実行させる、または、ネットワークを介して流通させることが可能である。

【 0 1 0 1 】

本発明は、上記の実施の形態に限定されることなく、特許請求の範囲内において、種々変更・応用が可能である。

【符号の説明】

【 0 1 0 2 】

- 1 IPネットワーク
- 10 object
- 20 リゾルバ

10

20

30

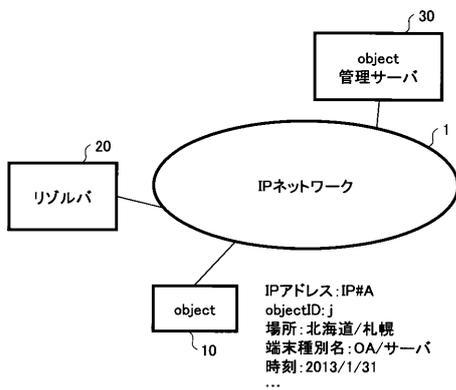
40

50

- 2 1 登録・検索要求受付部
- 2 2 object名前付与・クエリベクトル生成部
- 3 0 object管理サーバ
- 3 1 object・クエリベクトル受付部
- 3 2 object登録・検索部
- 3 3 objectデータベース
- 3 4 ツリー構築部

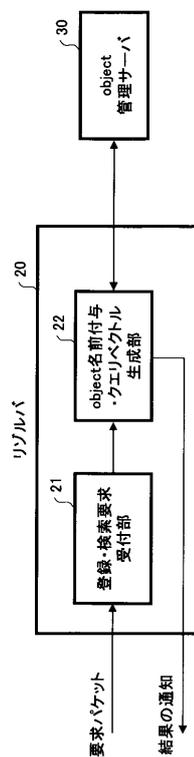
【図1】

本発明が適用されるIPネットワークの基本構成例



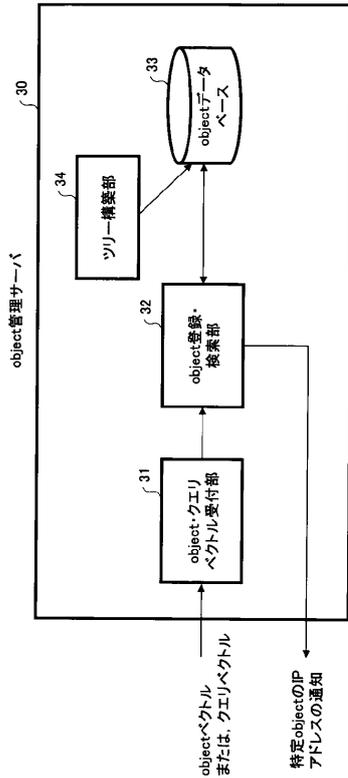
【図2】

本発明のリゾルバの構成例



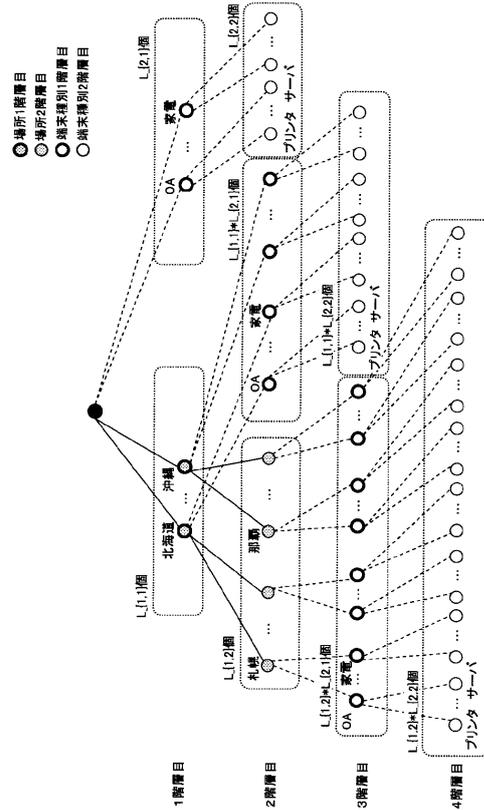
【 図 3 】

本発明のobject管理サーバの構成例



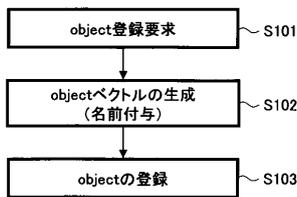
【 図 4 】

本発明の第1の実施の形態における組み合わせ管理ツリーの例



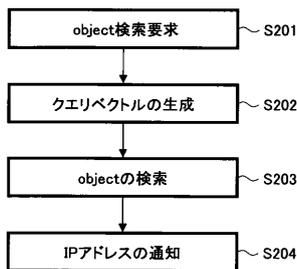
【 図 5 】

本発明の第1の実施の形態における名前付与・登録時のフローチャート



【 図 6 】

本発明の第1の実施の形態における検索(名前解決)時のフローチャート



---

フロントページの続き

審査官 田中 秀樹

(56)参考文献 特開2001-195414(JP,A)

特開2002-117039(JP,A)

川原 亮一, 外1名, 実世界情報を用いた複数名前付けと名前解決方法, 電子情報通信学会2013年総合大会講演論文集 通信2, 日本, 電子情報通信学会, 2013年 3月 5日, P.197

(58)調査した分野(Int.Cl., DB名)

G06F 17/30