

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4695759号
(P4695759)

(45) 発行日 平成23年6月8日(2011.6.8)

(24) 登録日 平成23年3月4日(2011.3.4)

(51) Int.Cl.

F 1

G06F 12/00	(2006.01)	G06F 12/00	546M
G06F 13/00	(2006.01)	G06F 13/00	520C
G06F 17/30	(2006.01)	G06F 17/30	110C

請求項の数 18 (全 23 頁)

(21) 出願番号	特願2000-560513 (P2000-560513)
(86) (22) 出願日	平成11年7月14日 (1999.7.14)
(65) 公表番号	特表2002-520735 (P2002-520735A)
(43) 公表日	平成14年7月9日 (2002.7.9)
(86) 國際出願番号	PCT/US1999/015951
(87) 國際公開番号	W02000/004458
(87) 國際公開日	平成12年1月27日 (2000.1.27)
審査請求日	平成16年6月25日 (2004.6.25)
審判番号	不服2008-24027 (P2008-24027/J1)
審判請求日	平成20年9月18日 (2008.9.18)
(31) 優先権主張番号	60/092,710
(32) 優先日	平成10年7月14日 (1998.7.14)
(33) 優先権主張国	米国 (US)
(31) 優先権主張番号	09/314,863
(32) 優先日	平成11年5月19日 (1999.5.19)
(33) 優先権主張国	米国 (US)

(73) 特許権者	596004587 マサチューセッツ インスチチュート オブ テクノロジー アメリカ合衆国 O 2 1 3 9 マサチューセッツ州, ケンブリッジ, マサチューセツ アベニュー 77
(74) 代理人	100066692 弁理士 浅村 皓
(74) 代理人	100072040 弁理士 浅村 肇
(74) 代理人	100094673 弁理士 林 銘三
(74) 代理人	100107467 弁理士 員見 正文

最終頁に続く

(54) 【発明の名称】組込型コンテンツ分散ゴースト・サーバを用いたグローバル文書ホスティング・システム

(57) 【特許請求の範囲】

【請求項 1】

クライアント・マシンがコンテンツ・プロバイダ・サーバにコンピュータ・ネットワークを介して接続される分散型ホスティングシステムであって、

ウェブ・ページの少なくとも 1 つの組み込みオブジェクト URL のドメイン名とパスに前記コンテンツ・プロバイダ・サーバとは異なる一組のコンテンツ・サーバのホスト名を追加するように、前記ウェブ・ページの少なくとも 1 つの組み込みオブジェクト URL を修正変更する手段を有し、該修正変更する手段によって修正変更された URL を含む前記ウェブ・ページを前記クライアント・マシンへ送信する前記コンテンツ・プロバイダ・サーバと；

前記コンテンツ・プロバイダ・サーバとは異なる一組のコンテンツ・サーバであって、前記コンテンツ・プロバイダ・サーバにより通常管理されているウェブ・ページの少なくとも 1 つの組み込みオブジェクトを担当する前記一組のコンテンツ・サーバと；

前記クライアント・マシンから前記コンテンツ・プロバイダ・サーバとは異なる一組のコンテンツ・サーバのホスト名を受信し、複数の第二レベル DNS サーバのうちのいずれかの IP アドレスを返送する第一レベル DNS サーバと；

前記クライアント・マシンから前記コンテンツ・プロバイダ・サーバとは異なる一組のコンテンツ・サーバのホスト名を受信し、前記コンテンツ・プロバイダ・サーバとは異なる一組のコンテンツ・サーバのうちのいずれかの IP アドレスを返送する前記複数の第二レベル DNS サーバとを含み；

10

20

前記クライアント・マシンから発せられたウェブ・ページに対する要求に応じて、修正変更された組み込みオブジェクトURLを含む前記ウェブ・ページが前記コンテンツ・プロバイダ・サーバからサービスされ、修正変更された組み込みオブジェクトURLによって識別された前記組み込みオブジェクトが、前記第一レベルおよび前記第二レベルDNSサーバで識別された前記コンテンツ・サーバの1つからサービスされる、前記ホスティングシステム。

【請求項2】

請求項1記載のホスティングシステムであって、冗長な第一レベルDNSサーバを更に含む、前記ホスティングシステム。

【請求項3】

請求項1記載のホスティングシステムであって、冗長な第二レベルDNSサーバを更に含む、前記ホスティングシステム。

【請求項4】

請求項1記載のホスティングシステムにおいて、前記一組のコンテンツ・サーバのうちの1つは、前記一組のコンテンツ・サーバのうちの前記1つが故障状態となった際のホスティング責任を肩代わりするための1つのバディー・サーバを含む、前記ホスティングシステム。

【請求項5】

請求項1記載のホスティングシステムにおいて、前記第二レベルDNSサーバが、前記一組のコンテンツ・サーバの部分集合の中で負荷をバランスさせる、負荷バランス機構を含む、前記ホスティングシステム。

【請求項6】

請求項5記載のホスティングシステムにおいて、前記負荷バランス機構が前記一組のコンテンツ・サーバのいずれの容量も超えることなく、組み込みオブジェクトに対して要求された複写量を最小化する、前記ホスティングシステム。

【請求項7】

請求項1記載のホスティングシステムが更に、前記一組のコンテンツ・サーバの指定された部分集合の容量を超えることなく、クライアント・マシンで経験される待ち時間の総量を最小化するための、オーバ・フロー制御機構を含む、前記ホスティングシステム。

【請求項8】

請求項7記載のホスティングシステムにおいて、前記オーバ・フロー制御機構が最小コスト多重利益フロー・アルゴリズムを含む、前記ホスティングシステム。

【請求項9】

請求項1記載のホスティングシステムにおいて、前記第一レベルDNSサーバが、要求を前記組み込みオブジェクトに向かわせる際に使用される、ネットワーク・マップを含む、前記ホスティングシステム。

【請求項10】

請求項1記載のホスティングシステムにおいて、前記一組のコンテンツ・サーバの1つが、指定された組み込みオブジェクトに対する総トラヒックを指定された制限内に維持するための、ゲート機構を含む、前記ホスティングシステム。

【請求項11】

請求項10記載のホスティングシステムにおいて、前記ゲート機構が：

指定された前記組み込みオブジェクトに対する要求数が、指定された閾値を超えたか否かを判定する手段と；

前記判定手段に応答し、指定された前記組み込みオブジェクトのサービスを制限するための手段とを含む、前記ホスティングシステム。

【請求項12】

請求項11記載のホスティングシステムにおいて、前記制限するための手段が、指定された前記組み込みオブジェクトより小さなオブジェクトをサービスするための手段を含む、前記ホスティングシステム。

10

20

30

40

50

【請求項 1 3】

請求項 1 1 記載のホスティングシステムにおいて、前記オブジェクトが、クライアントが指定された組み込みオブジェクトを後で受信することを可能とする切符である、前記ホスティングシステム。

【請求項 1 4】

U R L によりそれぞれ識別される組み込みオブジェクトを含むマークアップ言語に基づくドキュメントを有するページをコンテンツ・プロバイダ・サーバによりサービスする方法において、

前記コンテンツ・プロバイダ・サーバ上でコンピュータプログラムを実行することにより、組み込みオブジェクトのキャッシュされたコピーが利用できないときに何時でも前記組み込みオブジェクトをリトリープする際に使用するために修正変更された U R L の一部として保守され、元のホスト名に追加された、新たなホスト名を含む修正変更された U R L を生成するように前記組み込みオブジェクトの前記 U R L を書き換えるステップと、
10

前記コンテンツ・プロバイダ・サーバを用いることにより、コンテンツ・プロバイダ・サイトが受信したページに対するサービスの要求に応答し、前記修正変更された U R L を含む前記ページをサービスするステップと、

前記新たなホスト名により識別される前記コンテンツ・プロバイダ・サーバ以外のコンテンツ・サーバから前記組み込みオブジェクトをサービスするよう試行するステップと、

前記コンテンツ・サーバから前記組み込みオブジェクトが利用できないときに前記組み込みオブジェクトを前記コンテンツ・プロバイダ・サーバからサービスするステップを含む、前記方法。
20

【請求項 1 5】

ページと、ページオブジェクトをサービスするための方法であって、前記ページはコンテンツ・プロバイダ・サーバに格納され、前記ページオブジェクトのコピーは前記コンテンツ・プロバイダ・サーバとは異なる一組のコンテンツ・サーバに格納される前記サービスするための方法であって：

(a) 前記コンテンツ・プロバイダ・サーバ上でコンピュータプログラムを実行することにより、コンテンツ・プロバイダが供給したドメイン名とパスに追加されたホスト名を含むように前記ページオブジェクトのための U R L を修正変更するステップと、

(b) 前記修正変更された U R L を含む前記ページを前記コンテンツ・プロバイダ・サーバからサービスするステップと、
30

(c) ネーム・サーバを用いて、前記ホスト名を解読するためのブラウザからの要求に応答して、オブジェクトをリトリープするように前記一組のコンテンツ・サーバの 1 つのコンテンツ・サーバを識別するステップと、

(d) 前記ブラウザに前記 1 つのコンテンツ・サーバからの前記オブジェクトをリトリープするよう前記識別されたコンテンツ・サーバの I P アドレスを前記ネーム・サーバから前記ブラウザに返却するステップとを含む、前記方法。

【請求項 1 6】

請求項 1 5 記載の方法において、前記ページオブジェクトの前記コピーは前記一組のコンテンツ・サーバの部分集合に格納される、前記方法。
40

【請求項 1 7】

コンテンツを配信する方法であって、

ページ内の組み込みオブジェクトの U R L をコンテンツ・プロバイダ・ドメイン以外のドメインに解読するため、コンテンツ・プロバイダ・サーバが、コンテンツ・プロバイダ・ドメインから供給される U R L にデータを追加することにより替わりの U R L を生成するステップと、

前記替わりの U R L を含む前記ページを前記コンテンツ・プロバイダ・サーバからサービスするステップと、

D N S サーバが、前記替わりの U R L を解読して前記コンテンツ・プロバイダ・ドメイン以外のドメイン内のコンテンツ・サーバを識別する解読ステップと、
50

前記識別されたコンテンツ・サーバから前記組み込みオブジェクトをサービスするステップと、を含む前記方法。

【請求項 18】

請求項 17 記載の方法において、

前記替わりの URL を解読するステップが、前記コンテンツ・サーバを識別するため、最新のインターネット通信条件を識別するデータとユーザー位置を使用するステップを含む、前記方法。

【発明の詳細な説明】

【0001】

(技術分野)

10

本発明は一般的にコンピュータ・ネットワーク内での情報検索 (retrieval) に関する。更に詳細には、本発明はインターネット・サービス・プロバイダ (ISP) およびインターネット・コンテンツ・プロバイダの問題を解決するための、インターネット上でのコンテンツのホスティングおよび配信の新規な方法に関する。

【0002】

(関連技術の説明)

ワールド・ワイド・ウェブ (WWW) はインターネットのマルチメディア情報検索システムである。ウェブ環境の中で、クライアント・マシンはハイパー伝送プロトコル (HTTP: Hypertext Transfer Protocol) を使用してウェブ・サーバへのトランザクションを処理する、これはファイル (例えば、テキスト、グラフィックス、画像、音響、ビデオ等) への使用者のアクセスを提供する既知のアプリケーション・プロトコルであり、ハイパーテキスト・マークアップ言語 (HTML) として知られている標準ページ記述言語を使用する。HTML は基本文書形式を提供し、開発者が別のサーバ及びファイルへの「リンク」を指定することを可能とする。インターネットの範疇において、サーバへのネットワーク経路は、ネットワーク接続を定義する特別シンタックスを有する、いわゆるユニフォーム・リソース・ロケータ (URL: Uniform Resource Locator) で識別される。HTML 互換ブラウザ (例えば、ネットスケープ・ナビゲータ (Netscape Navigator) またはマイクロソフト・インターネット・エクスプローラ (Microsoft Internet Explorer)) をクライアント・マシンで使用することは、URL 経由でのリンクの指定を含む。それに応答してクライアントはリンク内で識別されたサーバへ要求を行い、その返事として文書または HTML に基づいてフォーマットされたその他のオブジェクトを受信する。ウェブ・サーバで保持されている文書の集合をウェブ・サイトと呼ぶ場合もある。

20

【0003】

ウェブ・サーバに関する従来技術ではそのコンテンツを別のサーバへ反映保存 (mirror) することが良く知られている。実際、現在のところコンテンツ・プロバイダーがそのコンテンツを読者により近いところに配置するための唯一の方法は、そのウェブ・サイトのコピーを国内的にまた国際的に異なる場所にあるウェブ・ホスティング所に設置されている機械の上に構築することである。ウェブ・サイトのこれらのコピーはミラー・サイトとして知られている。不幸にしてミラー・サイトはコンテンツ・プロバイダに対して不必要的経済的また運営上の重荷を追わせ、それらは規模による経済効果をもたらさない。経済的には、1つの一次サイトと1つのミラー・サイトを具備するコンテンツ・プロバイダの全体コストは一次サイト単一のコストの2倍以上である。この追加コストは2つの要因による：(1) コンテンツ・プロバイは各々のミラー・サイトに関して別々のホスティング設備を契約しなければならない、そして(2) コンテンツ・プロバイはミラー・サイトの同期を維持するための追加のオーバーヘッド費用を負担しなければならない。

30

【0004】

ミラーリングに関する問題点を解決する努力の中で、例えばシスコ (Cisco)、レゾネイテ (Resonate)、ブライト・タイガー (Bright Tiger)、F5 ラブズ (F5 Labs) およびアルテオン (Alteon) の様な企業は、ミラー・サイトの同期と負荷のバランスを保たせるソフトウェア並びにハードウェアを開発している。これらの機械はコンテンツ・プロバイ

40

50

ダの助けになるが、内在している規模拡張可能性 (scalability) の問題解決には失敗している。コンテンツ・プロバイダはミラーリングに関するコストを負担する意志はあるが、この技術はそれ自体、数ウェブ（10未満）の規模を超えない。

【0005】

これらの経済的および規模拡張可能性の問題に加えて、ミラーリングはまた運用面での問題も残している。ミラー・サイトを使用するコンテンツ・プロバイダは遠隔地に物理的な空間を賃借りして管理しなければならないだけではなく、そのサイトを同期させかつ負荷バランスを取るためのソフトウェアおよびハードウェアを購入して保守しなければならない。現行の解決方法では、コンテンツ・プロバイダは複数のウェブ・サイトを保守するために必要な人員、技術およびその他の項目を供給する必要がある。纏めると、ミラーリングはコンテンツ・プロバイダに対してコンテンツを制作するという彼ら本来のビジネスに関係しない機能に対して経済的およびその他の資源を浪費させている。10

【0006】

更に、コンテンツ・プロバイダはまた彼らのコンテンツの管理を維持し続けたいと希望している。今日、ISPの中には、コンテンツ・プロバイダと最終使用者との間のリンクを中断できるキャッシュ・ハードウェアを設置しているところもある。この様なキャッシングの効果は破滅的な結果をコンテンツ・プロバイダにもたらし、それらには（1）コンテンツ・プロバイダが彼らのウェップ・ページへの正確なヒット数を得ることを妨げる（それによって広告主からの収入が減少する）、（2）コンテンツ・プロバイダがコンテンツを仕立て上げて特定の視聴者へ宣伝することを妨げる（これはコンテンツ・プロバイダのウェブ・ページの有効性を著しく制限する）、（3）時期遅れの情報をそのクライアントに提供することになる（これは最終使用者をイライラさせ怒らせることになる）というような事が含まれる。20

【0007】

当業分野において、使用者がインターネット・コンテンツを更に効率的に（すなわち不必要にネットワーク資源に負担を掛けることなく）獲得することを可能とし、同様にコンテンツ・プロバイダが彼らのコンテンツに対する管理を維持できるような、集中化されていないホスティング解法を提供する大きな必要性が残されている。

【0008】

本発明はこれらおよびその他の従来技術に関連する問題を解決する。30

【0009】

（発明の簡単な概要）

本発明の一般的な目的は、ウェブ・コンテンツを効率的に、効果的にかつ信頼性を持って最終使用者に提供するように設計された有機的で、大きな耐故障性を有する構造基板を形成する、多数の広域的に配備されたインターネット・サーバを含むコンピュータ・ネットワークを提供することである。

【0010】

本発明の別の更に一般的な目的は、ウェブ基盤コンテンツを配信するための基本的に新規でより良い方法を提供することである。本発明によるアーキテクチャは、好適に集中化された制御を具備しない分散型サーバの大規模ネットワーク上でコンテンツを知的に転送しかつ複写するための方法を提供する。40

【0011】

本発明の別の目的は、コンテンツを使用者の近くに移動させるネットワーク・アーキテクチャを提供することである。本発明によるアーキテクチャはウェブ・サイトが多数の視聴者の開発を、関連するトラヒックを取り扱うための大きな基盤構造を構築するとを気遣わずに行えるようにしている。

【0012】

本発明の更に別の目的は、分散型ウェブ・コンテンツ用の故障耐性のあるネットワークを提供することである。このネットワーク・アーキテクチャはより豊かなウェブ・ページの配信を高速化するために使用され、コンテンツ・プロバイダが多数の視聴者に対して信頼50

性高くまた経済的に、好適に最終使用者の近くに配置されているサーバからサービスを提供することが出来るようにしている。

【0013】

本発明の更に別の特徴は、コンテンツ・プロバイダとその最終使用者との直接的な関係に影響を与えることなく、大規模なネットワーク上でコンテンツを配信し管理出来ることである。

【0014】

本発明の更に別の特徴はインターネット用の分散型拡張可能基盤構造を提供することであり、これはウェブ・コンテンツを配信する負担をコンテンツ・プロバイダから好適に、例えば世界レベルで展開されている数百箇所のホスティング・サーバ・ネットワークにシフトさせる。 10

【0015】

一般的に本発明はネットワーク・アーキテクチャであり、これは真の全世界的規模でのホスティングを支援する。本発明による枠組み構造はコンテンツ・プロバイダがその最も人気のあるコンテンツを世界中の無制限のポイントに複写可能とする。別の特徴として、或る1つの地理的な場所で複写された実際のコンテンツがその場所での視聴者に特定的に調整される。更に、コンテンツはそれが要求された場所へ、コンテンツ・プロバイダ部分での努力またはオーバヘッドを伴うことなく自動的に送られる。

【0016】

従って本発明の更に一般的な目的は、コンテンツ・プロバイダが彼らのコンテンツの制御を維持することを可能とする、全世界的ホスティング枠組み構造を提供することである。 20

【0017】

本発明のホスティング枠組み構造は、分散方式で動作する1組のサーバを含む。サービス対象である実際のコンテンツは好適に1組のホスティング・サーバ(時にゴースト・サーバと呼ばれる)上でサポートされる。このコンテンツはH T M L ページ・オブジェクトを含み、これは従来通りコンテンツ・プロバイダ・サイトからサービスされる。しかしながら本発明によれば、ウェブ・ページの基本H T M L ドキュメントはコンテンツ・プロバイダのサイトからサービスされるが、そのページに対する1つまたは複数の組み込まれたオブジェクトはホスティング・サーバからサービスされ、好適にクライアントのマシンに最も近いホスティング・サーバからサービスされる。基本H T M L ドキュメントをコンテンツ・プロバイダのサイトからサービスすることにより、コンテンツ・プロバイダはそのコンテンツに対する管理を維持する。 30

【0018】

指定された組み込みオブジェクトのサービスを行うためにどのホスティング・サーバを使用するかの決定は、ホスティング枠組み構造内のその他の資源によって実施される。特にこの枠組み構造は第2組のサーバ(またはサーバ資源)を含み、これらは最上位レベルドメイン・ネーム・サービス(D N S)を提供するように構成されている。加えて、この枠組み構造はまた第3組のサーバ(またはサーバ資源)を含み、これらは低レベルD N S機能を提供するように構成されている。クライアント・マシンが指定されたウェブ・ページに対するH T T P 要求をウェブ・サイトに発すると、先に説明したように基本H M T L ドキュメントがウェブ・サイトからサービスされる。そのページに対する組み込みオブジェクトは、好適に最上位および低レベルD S N サーバで識別された特定のホスティング・サーバからサービスされる。使用する適切なホスティング・サーバの位置決めをするために、最上位レベルD N S サーバは、その組み込みオブジェクトに対する要求に応答するための指定された低レベルD N S サーバを識別するために、ネットワーク内での使用者の場所を決定する。続いて最上位レベルD N S サーバはその要求を識別された低レベルD N S サーバへ回送し、これは続いてその要求を、オブジェクトをそのクライアントに返すサービスを行う指定されたホスティング・サーバ用のI P アドレスに解読する。 40

【0019】

更に一般的に、いくつかのレベルから構成されたD N S サーバの階層を有することが可能 50

である（場合によっては好ましい）。階層の下に行くに従って、最適な地域に近づく。

【0020】

本発明の更に別の特徴は1つの手段であり、これによってコンテンツはサーバの集合を通して分配および複写する事が可能であり、メモリの使用がいくつかの制約を条件として最適化され、この制約とは要求を満足させるために全てのオブジェクトの十分な数のコピーが存在し、オブジェクトのコピーは分散されていてどのサーバも過負荷とならず、これらのコピーは時間と共に同一サーバ上に配置される傾向にあり、またそれらのコピーはそれらを要求するクライアントに近い地域に配置されるということである。従って、この枠組み構造の中で動作しているサーバはコンテンツ・データベース内の全てのコピーを保持するわけではない。むしろ、指定されたサーバは最小量のデータのコピーを保持し、全システムが要求された水準のサービスを提供するようにしている。本発明のこの特徴は、全てのものを全ての場所にキャッシングする技法や、またはオブジェクトを予め指定された場所のみにキャッシングする技法に比較して更に効率的なホスティング技法を可能とする。

10

【0021】

全世界的ホスティングの枠組み構造は、動作の各レベルにおいて故障耐性を有する。特に、最上位レベルDNSサーバは、組み込みオブジェクトに対する要求をサービスするためにクライアントによって使用されるはずの低レベルDNSサーバのリストを返す。同様に、各ホスティング・サーバは好適にバディーサーバを含み、これはその関連するホスティング・サーバが故障状態の際にホスティング責任を引き受けるために使用される。

20

【0022】

本発明によれば、ホスティング・サーバの組の間での負荷バランスは組み込みオブジェクト要求を分配するための新奇な技術によって一部実現されている。特に、各組み込みオブジェクトURLは好適に仮想サーバ・ホスト名をそのURLの中にプリpend（追加:prepend）することで修正変更される。更に一般的には仮想サーバ・ホスト名がURLの中に挿入される。好適に、この仮想サーバ・ホスト名は1つの値（シリアル番号と呼ばれる場合もある）を含み、この値は指定されたハッシュ関数をURLに適用したり、またはそのオブジェクトに関する情報をその値の中に符号化する事によって生成される。この機能は組み込みオブジェクトを仮想サーバ・ホスト名の指定された組の中にランダムに分散させるように働く。加えて、組み込みオブジェクトに対する指定された指紋値が、指定されたハッシュ関数を組み込みオブジェクト自体に適用して生成される。この指定された値はその組み込みオブジェクトが既に修正されているか否かを識別する指紋として働く。好適にそれらの値（すなわち、仮想サーバ・ホスト名および指紋用に）を生成するために使用される機能は、指定されたウェブ・ページに対してオフライン過程で適用される。従って、そのページに対するHTTP要求が受信されると、その基本HTMLドキュメントがそのウェブ・サイトでサービスされ、またそのページの組み込みオブジェクトのある部分がその要求を発したクライアント・マシンに近い（最も近い必要は無いが）ホスティング・サーバからサービスされる。

30

【0023】

以上で本発明の関連する目的並びに機能の概要を説明した。これらの目的は本発明の更に突出した機能およびアプリケーションのいくつかを単に図示するものと考えられるべきである。多くのその他の有益な結果を、開示された発明を異なる方法で適用したり、または本発明を記述されるように修正変更することによって得ることが出来る。従って、その他の目的および本発明の更に完全な理解は、好適な実施例の以下の詳細な説明を参照する事により得られるであろう。

40

【0024】

本発明およびその特長を更に完全に理解するために、添付図と関連してなされる以下の詳細な説明を参照する。

【0025】

（好適な実施例の詳細な説明）

既知のインターネット・クライアント・サーバ・システムが図1に図示されるように実現

50

されている。クライアント・マシン10がウェブ・サーバ12にネットワーク14を介して接続されている。図示する目的で、ネットワーク14はインターネット、イントラネット、イクストラネットまたは任意のその他の既知のネットワークである。ウェブ・サーバ12はクライアントでアクセス可能な複数のサーバの1つであって、クライアントの1つはマシン10で図示されている。代表的なクライアント・マシンはブラウザ16を含み、これはネットワークのサーバにアクセスするために使用される既知のソフトウェア・ツールである。ウェブ・サーバは複数のファイル(集合的に「ウェブ」サイトと呼ぶ)をハイパーテキスト・ドキュメントおよびオブジェクトの形式でサポートする。インターネットの世界ではサーバへのネットワーク・パスはいわゆるユニフォーム・リソース・ロケータ(URL)で識別される。

10

【0026】

代表的なウェブ・サーバ12はプロセッサ18、オペレーティング・システム20、およびネットスケープ・エンタプライズ・サーバ(Netscape Enterprise Server)の様なウェブ・サーバ・プログラム22を含むコンピュータである。サーバ12はまた管理および運用用のグラフィカル・ユーザ・インターフェース(GUI)、およびアプリケーション・プログラミング・インターフェース(API)サポートする表示器を含み、このアプリケーション・プログラミング・インターフェースはアプリケーション開発者に対して、その中枢機能の拡張そして/またはカスタマイズをコモン・ゲートウェイ・インターフェース(CGI)プログラム、プラグイン、サーブレット(Servlets)、アクティブ・サーバ・ページ(Active Server Pages)、サーバ・サイド・インクルード(SSI: Server Side Include)機能または相当機能を含むソフトウェア・プログラムを通して行えるようにしている。

20

【0027】

代表的ウェブ・クライアントはx86-、PowerPC^(R)またはRISC型のパーソナル・コンピュータであり、これはIBM(R)OS/2(R)またはMicrosoft Windows'95の様なオペレーティング・システムを含み、これはNetscape Navigator 4.0(またはそれ以上)の様な、Java Virtual Machine(JVM)を有しアプリケーション・プラグイン(application plugins)またはヘルパー・アプリケーション(helper application)をサポートするウェブ・ブラウザを含む。クライアントはまたノート型コンピュータ、携帯型コンピュータ装置(例えばPDA)、インターネット・アプライアンス(Internet appliance)またはその他のコンピュータ・ネットワークに接続可能な装置であっても構わない。

30

【0028】

図2に示されるように、典型的なウェブ・ページはマークアップ言語(例えばHTML)主または基本ドキュメント28、並びに多数の組み込みオブジェクト(例えば画像、音声、ビデオなど)30を含む。従って、典型的なページでは、20またはそれ以上の組み込み画像またはオブジェクトは非常に一般的である。これらの画像の各々はウェブ内で独立なオブジェクトであり、個別にリトリープ(または変更する事が認められている)されるものである。従ってウェブ・クライアントの一般的な行動様式は、基本HTMLドキュメントをフェッチし、続いて直ちに組み込みオブジェクトをフェッチする事であり、これらは典型的に(常にというわけでは無いが)同一サーバ上に配置されている。本発明によれば、好適にマークアップ言語基本ドキュメント28はウェブ・サーバ(すなわちコンテンツ・プロバイダ・サイト)からサービスされる一方で、指定された数(またはおそらく全て)の組み込みオブジェクトはその他のサーバからサービスされる。見て分かるように好適に指定された組み込みオブジェクトはクライアント・マシンに近く過負荷では無く、また要求されたファイルの最新版を既に持っていると思われるサーバ(ウェブ・サーバ自体以外)からサービスされる。

40

【0029】

次に図3を参照すると、この動作は本発明のホスティング・システムで実現される。見て分かるように、ホスティング・システム35は1組の広範に配備された複数のサーバ(またはサーバ資源)を含み、これは大規模、故障耐性を有する基盤構造を形成し、これはウェブ・コンテンツを効率的、効果的または信頼性高く最終使用者にサービスするように設

50

計されている。これらのサーバは世界的に配備されていても、また任意の希望する地理的地域に広がっていても良い。理解されるように、ホスティング・システムはその様なコンテンツを知的に回送しかつ複写するための分散型アーキテクチャを提供する。この目的のために、全世界的ホスティング・システム 35 は 3 つの基本型サーバ（またはサーバ資源）：ホスティング・サーバ（場合によってはゴーストと呼ばれる）36、最上位レベル DNS サーバ 38、および低レベル DNS サーバ 40 を含む。図示はされていないが、DNS 階層の中で別のレベルも存在するはずである。これに代わって、最上位レベルと低レベルサーバの機能を結合する単一 DNS レベルであっても構わない。この図示された実施例では、発明の枠組み構造 35 はインターネット・サービス・プロバイダ（ISP）で配備されているが、これは本発明の制約ではない。本発明の全世界的ホスティング枠組み構造 35 を配備する、1つまたは複数の ISP は好適に多数のマシンを有し、これらはそれらのネットワーク上のゴースト・サーバ構成要素 36 および低レベル DNS 構成要素 40 の両方を実行する。これらのマシンはネットワークを通して分散されており；好適にこれらはネットワーク交換ポイント 42 およびネットワーク・アクセス・ポイント 44 の周りに集中されているが、これは必要事項では無い。加えて、ISP は好適に最上位レベル DNS 38 で動作する少数のマシンを有し、これはまたネットワーク上に分散されている。10

【0030】

制約を意味しないが、好適に枠組み構造 35 の中で使用されている指定されたサーバは、プロセッサ、オペレーティング・システム（例えば、Linux, UNIX, Windows NT, 等）、ウェブ・サーバ・アプリケーション、及び本発明で使用される 1 組のアプリケーション処理ルーチンを含む。これらの処理ルーチンはプロセッサで実行される 1 組の命令としてソフトウェアで好便に実現されており、以下に更に詳細に説明する種々の処理工程または方法ステップを実施する。これらのサーバは好適にネットワークの先端（例えば、ポイント・オブ・プレゼンス、またはPOSSs）に配置されている。20

【0031】

ネットワーク内の何処にホスティング・サーバを設置するかを決定するにはいくつかの要因がある。従って、例えばサーバ位置は好適に需要主導ネットワーク・マップで決定され、これはプロバイダ（例えば ISP）がトラヒック要求を監視することを可能とする。トラヒック・パターンを研究する事により、ISP は与えられたトラヒック・プロフィールに対してサーバ位置を最適化する。30

【0032】

本発明によれば、指定されたウェブ・ページ（基本 HTML ドキュメントおよび 1 組の組み込みオブジェクトを含む）は分散方式でサービスされる。従って、好適に基本 HTML ドキュメントは通常そのページを管理するコンテンツ・プロバイダからサービスされる。組み込みオブジェクトまたはそのいくつかのサブセットは優先的にホスティング・サーバ 36 からサービスされ、特に最初にそのウェブ・ページに対する要求を発したクライアント・マシンに近い、指定されたホスティング・サーバ 36 からサービスされる。加えて、ホスティング・サーバ間で好適に負荷がバランスされ、クライアントがそのページを完成するそのオブジェクトを要求した際に、指定された組み込みオブジェクトがそのクライアント近くの指定されたホスティング・サーバから効率的にサービスされるようにしている。40

【0033】

この方法でページ・コンテンツをサービスするために、組み込みオブジェクトに関連する URL が修正変更される。良く知られているように、1 つのページ内でサービスされる各々の組み込みオブジェクトはそれ自身の URL を有する。典型的に URL は 1 つのホスト名を有し、これはこのオブジェクトが従来同様に、すなわち本発明を参照することなくサービスされるコンテンツ・プロバイダのサイトを識別する。本発明によれば、組み込みオブジェクト URL は最初に、好適にオンライン状態でその URL が全世界的ホスティング・サーバでサービスされるべく調整されるように修正変更される。オブジェクト URL を修正変更するための好適な方法を図示する流れ図が図 4 に図示されている。50

【 0 0 3 4 】

処理ルーチンはステップ 5 0 で開始され、指定されたページ内の全ての組み込みオブジェクトの処理が完了したか否かが判定される。全て完了している場合は、処理ルーチンは終了する。しかしながらそうで無い場合、処理ルーチンは次の組み込みオブジェクトをステップ 5 2 で獲得する。ステップ 5 4 で、仮想サーバ・ホスト名が指定された組み込みオブジェクトに対して U R L の中にプリペンドされる。この仮想サーバ・ホスト名は 1 つの値（例えば、数字）を含み、これは例えば指定されたハッシュ関数をその U R L に対して適用して生成される。良く知られているように、ハッシュ関数は任意長のビット・ストリングを入力として取り、固定長のビット・ストリング（ハッシュ値）を出力として生成する。この様な関数は 2 つの条件を満足する：（1）同一ハッシュ値を生成する 2 つの異なる入力を見つけることは不可能である、また（2）入力とそのハッシュ値を与えられた場合、同じハッシュ値を具備した異なる入力を見つけることは不可能である。ステップ 5 4 で、その組み込みオブジェクトに対する U R L が値 xx,xxx にハッシュされ、これは続いて仮想サーバ・ホスト名の中に含まれる。このステップはランダムにそのオブジェクトを指定された仮想サーバ・ホスト名に分配する。

10

【 0 0 3 5 】

本発明は、先に説明したハッシュ関数を適用することによる仮想サーバ・ホスト名の生成に限定するものでは無い。代わりのまた好適な実施例として、仮想サーバ・ホスト名が下記のように生成される。仮の見本的なホスト名 a1234.g.akamaitech_net を考える。1234 という値は、シリアル番号と呼ばれる場合もあり、好適にそのオブジェクトに関する情報、例えばそのサイズ（大きいかまたは小さいか）、その期待される人気、そのオブジェクトが生成された日時、ウェブ・サイトの識別子、オブジェクトの型式（例えば、動画または静止画）、およびおそらくは指定された乱数関数で生成されたいくつかの線数ビットを含む。もちろん、指定されたシリアル番号がその様な情報を全て符号化する必要もまたその様な構成要素の多くを符号化する必要も無い。実際、最も簡単な場合、このシリアル番号は单一整数であっても構わない。とにかく、情報は 1 つのシリアル番号の中に好適な方法で符号化される。従って、例えば第 1 ビットはサイズを表すために使用され、第 2 ビットは人気を表すために使用され、その他のビットの組は日時を表すために使用される等々。ハッシングの例のところで先に説明したように、シリアル番号はまた負荷バランス用に、また或る型式のトラヒックを或る型式のサーバに向けるためにも使用される。典型的に同一ページ上のほとんどの U R L は同一シリアル番号を有して、ページ毎に必要な識別名（DN : distinguish name）アクセス数を最小化する。この要求はオブジェクトが大きくなるに伴って重要さは少なくなる。

20

【 0 0 3 6 】

従って、本発明によれば仮想サーバ・ホスト名が指定された組み込みオブジェクトに対して U R L の中にプリペンドされ、このホスト名は 1 つの値（またはシリアル番号）を含み、これは指定された関数をその U R L またはオブジェクトに適用して生成される。この関数はハッシュ関数、符号化関数等である。

30

【 0 0 3 7 】

流れ図に戻ると、処理ルーチンは次にステップ 5 6 に続き、指定された値をオブジェクトの U R L に含める。好適に、指定された値は指定されたハッシュ関数を組み込みオブジェクトに適用して生成される。このステップはそのオブジェクトのユニークな指紋を生成し、これはそのオブジェクトが修正変更されているか否かを判定するために有用である。その後、この処理ルーチンはステップ 5 0 に戻って繰り返される。

40

【 0 0 3 8 】

以上の背景の下に、本発明の全世界的ホスティング枠組み構造を特定の例を参照して次に説明する。特に、ボストンのクライアント・マシンの使用者が、通常アトランタで管理されているコンテンツ・プロバイダのウェブ・ページを要求すると仮定する。図示を目的として、コンテンツ・プロバイダがネットワーク内で全世界的ホスティング・アーキテクチャを使用していると仮定する、このネットワークは全世界的、国際的、国内、地域、ロー

50

カルまたは私的なものであって構わない。図5はシステムの種々の構成要素とクライアントからの要求が如何にして処理されるかを示している。先に説明したようにこの動作は制約を目的としているものではない。

【0039】

ステップ1：ブラウザが要求をプロバイダのウェブ・サイトに送る（項目1）。アトランタのコンテンツ・プロバイダ・サイトは全世界的ホスティング枠組み構造が実施されていない場合と同様にしてこの要求を受信する。違いは何がプロバイダ・サイトから返却されるかである。通常のページを返却する代わりに、本発明に基づいてこのウェブ・サイトは図4の流れ図に図示されている方法に基づき修正変更された組み込みオブジェクトURLを具備したページを返却する。先に説明したように、これらのURLは以下のように変更される：

【0040】

100,000個の仮想ゴースト・サーバが存在すると仮定する、もっともネットワーク上には比較的少数（例えば100）しか物理的には存在しないかも知れないが。これらの仮想ゴースト・サーバまたは仮想ゴーストはホスト名で識別される：ghostxxxx.ghosting.com ここでxxxxは0から99,999の間の数で置換される。コンテンツ・プロバイダのウェブ・サイトが新たな情報で更新された後、コンテンツ・プロバイダ・サイト上で実行するスクリプトが実行されて、組み込みURLを書き換える。好適に、組み込みURL名は0と99,999の間の数にハッシュされる、もっともこの範囲は本発明を制限するものではない。次に組み込みURLがその数を具備する仮想ゴーストを参照するように切り換えられる。例えば、下記はプロバイダのサイトからの組み込みURLである：

【0041】

このURLで参照されたオブジェクトに対するシリアル番号が数1467の場合、好適にこのURLは次のように書き換えられる：

【0042】

30 この方法でシリアル番号を使用することにより、組み込みURLが100,000個の仮想ゴースト・サーバ名に対してほぼ均等に分散される。プロバイダ・サイトは個別の好みに応じて、スクリーン上の種々のオブジェクトを再配置してそのページを個人化出来ることに注意されたい。更に、プロバイダはまた広告をダイナミックに挿入して、どれだけの人が各々の広告を見たかを計数する事が可能である。

【0043】

好適な実施例に基づけば、組み込みURLに対して追加の修正変更がなされて、全世界的ホスティング・システムがサービスする情報が陳腐化しないように保証している。先に説明したように、好適に組み込みURLに含まれるデータのハッシュもまた組み込みURL自身の中に挿入されている。すなわち、各々の組み込みURLはそれが指示するデータの指紋を含んでいる。含まれる情報が変化すると、指紋も変化し、これによって使用者が古いデータを参照する事が防止される。

【0044】

第2ハッシュはビットのストリームを入力として取り込み、そのストリームの指紋と呼ばれることがある出力を生成する。指紋の重要な属性は、2つの異なるストリームはほぼ確実に2つの異なる指紋を生成することである。その様なハッシュの例はMD2およびMD3関数であるが、例えば単純なチェックサムの様な更に透過性の良い方法も使用できる。具体化するために、ハッシュの出力を128ビット署名と仮定する。この署名は1つの数字として解釈され、続いて組み込みURLの中に挿入される。例えば、プロバイダ・ウェブ・サイトからの画像space.story.gif内のデータのハッシュが数28765の場合、修正変

10

20

30

40

50

更された組み込みURLは実際下記のように見えるはずである：

<IMG

SRC=http://ghost1467.ghosting.akamai.com/28765/www.provider.com/TECH/images/space.story.gif">

【0045】

ページが変更される度に、好適に各々の組み込みURLに対するハッシュが再計算され、必要に応じてそのURLが書き換えられる。URLのいずれかのデータが変化する場合、例えば新たなまた異なる画像がspace.story.gifの名前で挿入される場合、そのデータのハッシュは異なるのでURL自身が別のものとなる。この手法は元のページが更新された結果、陳腐化されたデータをこのシステムがサービスすることを防止する。

10

【0046】

例えば、画像space.story.gifがコンテンツ・プロバイダのサーバ上で更に最新の版に置き換えられたと仮定する。画像のデータが変化するので、URLのハッシュも同様に変化する。従って、新たな組み込みURLは新たな数が指紋に挿入されない限り同一に見える。更新の後にそのページを要求した全ての使用者は、新たな画像を指示するページを受信する。旧い画像は決して参照されることではなく、更に更新された情報の代わりに間違って返却することは出来ない。

【0047】

纏めると、好適に2つのハッシュ操作が存在し、コンテンツ・プロバイダのページを修正変更するために行われる。第1に、ハッシングはドメイン名を仮想ゴースト名に変換するためのシリアル番号が選択される処理工程の、1つの構成要素である。理解されるように、この第1変換はクライアントを全世界ホスティング・システムに転送して組み込みURLをリトリープせしめるように働く。次に組み込みURLで指示されたデータのハッシュが計算されそのURLの中に挿入される。この第2変換は陳腐化および時期遅れのコンテンツがゴースト・サーバからサービスされることを防止する。好適に、これら2つの変換はオフラインで実施されるので性能上の隘路となる可能性は無い。

20

【0048】

一般化すると好適なURL技法は次の通りである。図示されているドメイン www.domainname.com/frontpage.jpg は次のように変換される：

30

xxx.yy.zzzz.net/aaaa/ www.domainname.com/frontpage.jpg,

此処で

xxxx=シリアル番号フィールド

yy =低レベルDNSフィールド

zzzz=最上位レベルDNSフィールド

aaaa=その他の情報(例えば、指紋)フィールド

DNS階層の別のレベルが使用されている場合、別の低レベルDNSフィールド、例えば xxxx.y₁y₂.zzz.net/aaaa/.... が存在するはずである。

【0049】

ステップ2：初期ページをコンテンツ・プロバイダ・サイトから受信した後、ブラウザはそのページを表示するために組み込みURLをロードする必要がある。これを実施するための第1ステップは、変更されたホスト名を解明するために使用者マシン(または使用者ISPでの)DNSサーバに接触することである、この場合は：

40

ghost1467.ghosting.akamai.com である。理解されるように、本発明の全世界的ホスティング・アーキテクチャはDNSシステムを、その名前がクライアントに近くまたそのページを既に持っていると思われるゴーストの1つに解明されるように取り扱う。これがどの様に行われるかを理解するために、以下にクライアントから発せられたDNSクエリー(query)の進行を説明する。

【0050】

ステップ3：先に説明したように、好適に2つの型式のDNSサーバが本発明のシステム

50

内に存在する：最上位レベルと低レベルである。ghosting.com用の最上位レベルD N S サーバ3 8は、.comドメインの通常のD N S サーバとは異なる特別の機能を有する。最上位レベルD N S サーバ3 8は適切な制御処理ルーチンを含み、これはネットワーク内の何処に使用者が配置されているかを決定し、続いてその使用者を近くの akamai.com（すなわち、低レベルD N S）サーバ4 0に転送するために使用される。.comドメインと同様に、akamai.comは好適にネットワークを通して分散されている多数の最上位レベルD N S サーバ3 8を、故障耐性を高めるために有している。従って指定された最上位レベルD N S サーバ3 8は使用者をインターネット（指定された組み込みオブジェクトに対する要求を満足するために使用されるホスティング・サーバ3 6の集合を有する）内の地域に転送し、一方低レベルD N S サーバ4 0（識別された地域内の）はその集合内で特定のホスティング・サーバを識別し、ここからオブジェクトが実際にサービスされる。

10

【 0 0 5 1 】

更に一般的に先に触れたように、D N S 処理はいくつかのレベルの処理を含むことが可能であり、その各々はクライアントをより良いゴースト・サーバに向けるように働く。ゴースト・サーバ名はまた更に多くのフィールドを持つことが可能である。例えば、“a123.g.ghost.akamatech.net”を“a123.ghost.akamai.com.”の代わりに使用することが出来る。唯ひとつD N S レベルのみが使用されている場合、代表U R Lは“a123.akamai.com.”である。

【 0 0 5 2 】

その他の技術を使用することも可能であるが、使用者のネットワーク内での場所は、要求を行ったクライアント・マシンのI P アドレスを調べることによって好適に演繹される。この例において、D N S サーバは使用者のマシーン上で動作しているが、これは要求事項ではない。使用者が例えばI S P D N S サーバを使用している場合、処理ルーチンはその使用者がこのサーバの近く（インターネットの意味で）に配置されていると仮定する。これに代わって、使用者の場所またはI P アドレスを最上位レベルD N S に送られた要求の中に直接符号化することが出来る。ネットワーク内のI P アドレスの物理的位置を決定するために、最上位D N S サーバはネットワーク・マップを構築し、これは続いて関連する位置を識別するために使用される。

20

【 0 0 5 3 】

従って、例えば最上位D N S にa1234.g.akamatech.net に関する解釈要求が来た際に、最上位D N S は要求者の返却アドレスを調べて、ネットワーク・マップに従い、そのアドレスに基づいて応答を作り出す。この例で、a 1 2 3 4 はシリアル番号、g は低レベルD N S を参照するフィールド、そしてakamatechは最上位レベルD N S を参照する。このネットワーク・マップは好適に全てのインターネット・プロトコル（I P ）ブロックのリストを含み、各々のI P ブロックに対して、マップはその要求を送信する場所を決定する。このマップは好適にネットワーク条件およびトラヒックに基づいて絶えず更新される。

30

【 0 0 5 4 】

ネットワーク内の何処で要求が発せられたかを決定した後、最上位D N S サーバはそのD N S 要求をネットワーク内でその使用者に近い低レベルD N S サーバに転送する。要求を転送する能力はD N S システム内の標準機能である。加えて、この転送はそのローカル低レベルD N S が停止している場合、接続されるバックアップ・サーバが存在するような方法で実施できる。

40

【 0 0 5 5 】

好適に、これらの最上位レベルD N S のghosting.comドメインに関する転送上のT T L (time to live)スタンプは長めに設定されている。これにより使用者のD N S サーバそして/またはI S P のD N S サーバでのD N S キャッシングが最上位レベルD N S サーバが過負荷状態となることを防止できる。使用者マシンまたはI S P のD N S サーバ内のghosting.akamai.comに対するT T L が経過すると、最上位レベル・サーバに接続され、ローカル低レベルghosting.akamai.comD N S サーバに対する新たな転送が、新たなT T L スタンプと共に返却される。このシステムは現在の集中化されたホスティング解釈の中で行わ

50

れているものに比較して、桁違いに多量の最上位レベル DNS 検索を行わしめるもので無いことに注意されたい。これは、最上位レベル転送の TTL が高く設定されているためであり、従ってほとんど多数の使用者は彼らのローカル DNS で、近くの低レベル ghosting.akamai.com DNS サーバに直接導かれるからである。

【 0 0 5 6 】

更に、最上位レベル DNS サーバに対する故障耐性は、一般的な .com ドメインで行われているのと同様に DNS によって自動的に提供される。低レベル DNS サーバに対する故障耐性は、好適に單一サーバのみの代わりに可能性のある低レベル DNS サーバのリストを返却することで提供される。低レベル DNS サーバの 1 つが停止していても、未だ使用者はリスト上に載っていて動作している 1 つに接続することが可能である。

10

【 0 0 5 7 】

故障耐性はまた「オーバフロー制御」機構を介して処理することも可能であり、ここでクライアントはそのオブジェクトにサービスを行う十分な容量を有することが知られている、地域内の低レベル DNS に転送される。この代替手法は特定の地域からの大量の要求が存在するとき、または 1 つの地域内の容量が縮退している様な状況で非常に有用である。一般的に、クライアントが地域に転送される方法は、どの地域も過負荷とならないという制約の下で、複数のクライアントが経験する総待ち時間を最小とするようになされる。地域容量の制約の下で総待ち時間を最小化することは、好適に最小コスト多重利益フロー・アルゴリズム (min-cost multicommodity flow algorithm) を使用して実現できる。

【 0 0 5 8 】

ステップ 4：この時点で、使用者は近隣の ghosting.com DNS サーバ 3 8 のアドレスを有する。使用者のローカル DNS サーバは近隣の低レベル DNS サーバ 4 0 に接続し、名前 ghost1467.ghosting.akamai.com への変換を要求する。ローカル DNS サーバは、使用者に近く、過負荷状態で無く、また要求されたデータを既に持っていると考えられる、ネットワーク上のゴースト・サーバ 3 6 の 1 つの IP アドレスを返却する責任がある。

20

【 0 0 5 9 】

仮想ゴースト名を実際のゴーストにマッピングするための基本機構はハッシングである。1 つの好適な技術はいわゆる無矛盾ハッシングであり、米国明細書 No.09/042,228、1998 年 3 月 13 日登録、米国明細書 No.09/088,825、1998 年 6 月 2 日、各々の名称は「複数の資源の中で要求を分配するための方法並びに装置」、マサチューセッツ工科大学所有、に記述されており、この明細書は此处に挙げることで参照されている。無矛盾ハッシュ関数はマシン故障およびクラッシュの際にシステムを強固なものとする。これはまたシステムをしなやかに、ほとんどの項目の配置を変更することなく、またこのシステムに関する完全な情報を持つことなく成長させることを可能とする。

30

【 0 0 6 0 】

本発明によれば、仮想ゴースト名は対照表を用いて実ゴースト・アドレスにハッシュされ、ここで対照表はネットワーク状態およびトラヒックに基づいて、負荷バランスと故障耐性が保証されるように、断続的に更新される。好適に、解読の表は各々のシリアル番号に対して生成される。例えば、シリアル番号 1 はゴースト 2 および 5 に解読される、シリアル番号 2 はゴースト 3 に解読され、シリアル番号 3 はゴースト 2, 3, 4 に解読される等々である。目的はどのゴーストもその容量を超えない解を定義し、全ての解の中で全てのゴーストの総数が最小化されるようにする事である。これはシステムが各々の地域で利用できるメモリの最大利益を確実に享受出来るようにされる。この点は既存の負荷バランス手法に較べて大きな特長であり、従来の手法は全てのものを全ての場所にキャッシュしたり、または負荷に関係なく或る場所の或るオブジェクトのみをキャッシュする傾向にある。一般的に、負荷が短い時間の間に余り大きく変化しない場合は、解が一貫して時間と共に変化しない様に割り当てられることが望ましい。この手法はまた好適に、ゴーストがどれだけ使用者に近い、またその時点でそのゴーストにどれだけ重い負荷が掛かっているかも考慮している。

40

【 0 0 6 1 】

50

同一の仮想ゴーストは好適に、その使用者がネットワーク内の何処に配置されているかによって異なる実ゴースト・アドレスに変換されることも注意されたい。例えば、ゴースト・サーバ18.98.0.17が米国内に配置されており、またゴースト・サーバ132.68.1.28はイスラエル国内に配置されていると仮定する。ボストンから発せられた `ghost1487.ghosting.akamai.com` に対する DNS 要求は、18.98.0.17に解読され、一方テル・アビブから発せられた要求は132.68.1.28に解読されるはずである。

【 0 0 6 2 】

2つのレベルの DNS サーバは種々のゴースト・サーバを監視してそれらの負荷を考慮に入れる一方、仮想ゴースト名を実アドレスに変換する。これはゴースト上および低レベル DNS サーバ上で動作するソフトウェア処理ルーチンで取り扱われる。1つ実施例において、負荷情報は1つの地域内のサーバ間で循環され、それらが各々のシリアル番号に対して解を計算できるようにしている。解を計算するための1つのアルゴリズムは下記のように働く。サーバは最初に各々のシリアル番号に対して計画された負荷（使用者要求の数に基づいて）を計算する。次にこのシリアル番号は負荷が増える順に処理される。各々のシリアル番号に対して、要求されるサーバのランダムな優先順位リストが無矛盾ハッシング法を用いて割り当てられる。次に各々のシリアル番号が優先順位リストからサーバの最小初期セグメントに解かれ、どのサーバも過負荷とならないようにされる。例えば、シリアル番号に対する優先順位リストが 2 , 5 , 3 , 1 , 6 の場合、最初にそのシリアル番号に対する負荷をゴースト 2 にマッピングすることを試みるように試行される。これがゴースト 2 を過負荷にする場合、その負荷はゴースト 2 および 5 の両方に割り当てられる。これによってこれらのサーバのいずれかに過大な負荷が掛かる場合、その負荷はゴースト 2 , 3 , および 5 といった具合に割り当てられる。1つのサーバ上での計画された負荷はそのサーバを含む全ての解を調べ、またそのシリアル番号からそのサーバに送られると予想される負荷の量を加えることで計算出来る。この解を生成する方法は、繰り返し形式で使用される際に最も効果的であり、そこでは割り当ては全てのシリアル番号が全てのゴーストにマッピングされているデフォルト状態から開始される。先の処理手順に基づいて解テーブルを洗練することにより、負荷は最小の繰り返し量を使用してバランスされる（これにより1つの地域内の利用可能なメモリが最大限節約される）。

【 0 0 6 3 】

これら低レベル DNS 変換に関する TTL は短く設定されていて、ゴーストの1つに重い負荷が検出された際に迅速に応答出来るようにしている。この TTL は1つのパラメータであって、ゴースト上の高負荷に対するタイムリーな応答と低レベル DNS サーバ上で誘発された負荷との間のバランスを保つために、システムで調整することが出来る。しかしながら、低レベル DNS 変換に対する TTL が 1 - 2 分に設定されていたとしても、実際に低レベル DNS 対照をしなければならない使用者はほとんど居ないことに注意されたい。ほとんどの使用者は彼らのマシンまたは彼らの ISP 上にキャッシュされている DNS 変換を見るはずである。従って、ほとんどの使用者は彼らのローカル DNS サーバから直接、彼らが望むデータを有する近隣のゴーストへ行く。実際に低レベル DNS 対照を行うそれらの使用者には非常に短い待ち時間が追加される、しかしながらこの待ち時間はほとんどのデータを近隣からリトリーブする長所を鑑みると短い。

【 0 0 6 4 】

先に示したように、低レベル DNS サーバに対する故障耐性は、最上位レベル DNS に対して単一のサーバ・アドレスの代わりに、可能な複数の低レベル DNS サーバのリストを返却することにより提供される。使用者の DNS システムはこのリストをキャッシュし（標準 DNS システムの一部）、リスト上の最初の1つが何らかの理由で停止している場合にはリスト上の別のサーバの1つに接続する。低レベル DNS サーバは DNS の標準機能を使用して、ゴースト・サーバに対して特別水準の故障耐性を提供する。名前が変換される際に、單一名を返却する代わりに、名前のリストが返却される。何らかの理由でゴーストに対する一次故障耐性方法（バディー・システム（Buddy system）として知られていて、後ほど説明する）が故障している場合、クライアントのブラウザはリスト上のその他の

10

20

30

40

50

ゴーストの1つに接続するはずである。

【0065】

ステップ5：次にブラウザはa123.ghosting.akamai.com/.../www.provider.com/TECH/images/space.story.gifと名付けられたオブジェクトに対して近隣のゴーストから要求を行う。元のサーバの名前(www.provider.com)が好適にURLの一部として含まれていることに注意されたい。ゴースト上で動作するソフトウェアはページ名を元のホスト名および実ページ名に分解する。ファイルのコピーが既にゴースト上に格納されている場合は、そのデータは直ちに戻される。しかしながら、データのコピーがゴースト上に存在しない場合は、コピーが元サーバまたは別のゴースト・サーバからリトリーブされる。ゴーストは元サーバが誰であるかを、その名前がブラウザからそのゴーストに送られてきたURLの中に符号化されているので知っていることに注意されたい。一度コピーがリトリーブされると、これは使用者に戻され、また好適にそれはまたゴースト上に要求に対して返答するために格納される。10

【0066】

追加の保証として、その使用者がそのサーバに本当に近いかを確認することが好適であろう。これはそのファイルに対する要求に応答する前に、クライアントのIPアドレスを調べて行われる。これはクライアントのDNSサーバがクライアントから遠く離れている様な希な場合に有用である。その様な場合、ゴースト・サーバはその使用者をより近いサーバ(または、そのクライアントにより近いサーバと解読されるに違いない別の仮想アドレス)に転送することが出来る。その転送が仮想サーバに対する場合、更に別の転送が行われるのを防止するために、目印しが付けられなければならない。好適な実施例では、転送は大きなオブジェクトに対してのみ実施される：従って、要求されたオブジェクトが或る総サイズを超えているかを確認するための確認が、転送を行う前になされる。20

【0067】

長いダウンロード性能は、変化するネットワーク条件に基づいて、クライアントが接続されているサーバをダイナミックに変更することにより改善できる。これはオーディオおよびビデオをダウンロードする際(ここでは接続は長くなりがちで品質が特に重要である)に特に有用である。この様な場合、使用者は流れ途中の別のサーバに向けられることが可能である。クライアントの転送を行うための制御構造は先に説明したものと類似しているが、クライアントのブラウザまたはメディア再生機の中に設置されるソフトウェアを含むことも出来る。このソフトウェアはクライアントの接続の性能およびおそらくはネットワークの状態も同様に監視する。クライアントの接続がサーバを変更することにより改善出来ると思われる場合は、システムはクライアントをその接続の残り部分に関して新たなサーバに向ける。30

【0068】

ゴーストに対する故障耐性は、相棒(バディー)・システムで提供され、此処では各々のゴーストが指定されたバディー・ゴーストを有する。1つのゴーストが停止すると、その相棒(バディー)がその仕事(およびIPアドレス)を引き継いで、サービスが中断されないようにする。このシステムの別の特徴はバディー・ゴーストが故障に対して待機時間を持たないことである。代わりに、全てのマシンは常に活動状態であり、故障が発生すると、その負荷はバディーによって肩代わりされ、低レベルDNSシステムにより別の動作中ゴーストとバランスが取られる。バディー・システムの別の特徴は、故障耐性が長い時間待つことなく提供されることである。40

【0069】

グローバル・ホスティング・システムの更に別の安全機能として、或るオブジェクトに対する総トラヒックを定められた制限内に保持するためにゲート機構が使用できる。ゲート機構の1つの実施例は次のように動作する。1つのオブジェクトに対する要求の数が、或る指定された閾値を超えると、そのサーバはそのオブジェクトをサービスしないように選択できる。これはそのオブジェクトが非常に大きな場合、非常に有効であろう。代わりに、そのクライアントはそのクライアントに後ほど再接続するように依頼するもっと小さな50

オブジェクトでサービスを受けることが出来る。または、そのクライアントを転送する事も可能である。ゲートを実現するための別の方法は、そのクライアントに対して「切符」を提供することであり、これはそのクライアントが予め定められた時間後にそのオブジェクトを受信できるようにするものである。この方法において、ゴースト・サーバはそのオブジェクトのサービスを行う前にその切符上の時刻を確認する必要がある。

【0070】

本発明のグローバル・ホスティング・システム手法はグローバルISPまたは地域ISPの集成体の方法であって、ホスティング売り上げを生み出すために彼らのネットワーク基盤構造を強化し、ネットワーク帯域幅を節約するためのものである。本発明のグローバル・ホスティング手法を提供するISPはコンテンツ・プロバイダに対してコンテンツをかれらの使用者に対してISPネットワーク上の最も近いポイントから配信する機能を提供し、従って高速で信頼性の高いアクセスを保証する。保証されたウェブ・サイト性能は全てのウェブを使用するビジネスに重要であり、グローバル・ホスティングはこの要求を満足するサービスを作り出すことを可能とする。

10

【0071】

本発明に基づくグローバル・ホスティングはまたISPがコンテンツがネットワークをどの様にまた何処を移動するかを制御することを可能とする。グローバル・ホスティング・サーバはISPネットワークの端点（例えば多数のネットワーク交換機およびアクセス・ポイント）に設定することが出来る。これはISPがそれが受け持っているサイトに対して、コンテンツを直接ネットワーク交換ポイントおよびアクセス・ポイントの中にサービスすることを可能とする。高価な基幹リンクがコンテンツ・プロバイダのサイトからネットワーク交換機およびアクセス・ポイントへ、冗長なトラヒックを搬送する必要はもはや無くなる。代わりに、コンテンツはISPのネットワークから、高価なネットワーク資源をその他のトラヒックに開放して、直接サービスされる。

20

【0072】

グローバル・ホスティングはネットワーク・トラヒックを低減するが、これはまたグローバルISPが急速に拡大しているホスティング市場の一部を獲得する手段であり、この市場拡大は現在毎年10億ドルと見込まれている。

【0073】

グローバル・ホスティング解法はまた多くの利益をコンテンツ・プロバイダにもたらし、特に彼らのウェブ・サイトの性能を国内的および国際的に改善する効果的でかつ経済的な解決策を提供する。本発明によるホスティング・ソフトウェアはコンテンツ・プロバイダに対して高速でかつ信頼性の高いインターネット・アクセスを、コンテンツを彼らの加入者に対してISPネットワーク上の最も近いポイントから配信する手段を提供することにより保証する。加えて、更に別の利点が更に詳細に以下に説明されており、グローバル・ホスティング解法はまたネットワーク・トラヒックを削減する重要な利点をも提供する。

30

【0074】

ひとたびグローバル・ホスティング・サーバがISPネットワークの周辺機器（すなわち、多数のネットワーク交換機およびアクセス・ポイント）にインストールされると、コンテンツはネットワーク交換機およびアクセス・ポイントの中に直接サービスされる。コンテンツを直接ISPのネットワークから効率的に配信する結果、本発明はウェブ・サイト性能を事実上改善する。現行のコンテンツ配信システムと比較して、本発明のグローバル・ホスティング解法は高価な基幹リンクを使用して冗長なトラヒックをコンテンツ・プロバイダのウェブ・サイトからネットワーク交換機およびアクセス・ポイントへ搬送する必要が無い。

40

【0075】

本発明のグローバル・ホスティング手法で提供される特有な特長を以下に示す：

【0076】

1. コンテンツ・プロバイダの運営費削減：

ほとんどの競合する解法はコンテンツ・プロバイダに対して彼らのコンテンツを担当する

50

各々のウェブ・サイトにサーバを購入することを要求する。その結果、コンテンツ・プロバイダは世界中で異なるISPと個別の契約を結ばなければならない。加えて、コンテンツ・プロバイダは一般的にこれらの遠隔場所でコンテンツの複写とサーバの保守の責任を負う。

【0077】

本発明では、ISPがグローバル・ホスティングの機能のほとんどの一次責任を負う。コンテンツ・プロバイダは好適に彼らの単一サーバのみを保守する。このサーバ上のコンテンツはソフトウェアによりそれらがアクセスされる場所へ自動的に複写される。プロバイダ（またはそれについてはISP）による介入または計画は不要である。コンテンツ・プロバイダにはグローバル・ネットワーク上の全てのサーバに対する瞬時のアクセスが提供される；コンテンツが複写されるべき場所の選択または遠隔地にサーバを追加購入する必要は無い。10

【0078】

2. インテリジェントかつ効率的データ複写：

ほとんどの競合する解法はコンテンツ・プロバイダに対して彼らのコンテンツを1つの商業的ホスティング・サイトのサーバ上に複写するか、または彼らのコンテンツを地理的に遠距離にあるサーバ上にミラー化するように要求する。どちらの方法も効率的ではない。前者の解法では、コンテンツは未だインターネット上の単一場所に配置されている（従って、ほとんどの使用者から遠く離れている）。後者の場合には、ウェブ・サイトの全てのコンテンツが遠隔サーバにコピーされるが、コンテンツの小さな部分であっても実際、遠隔地に配置される必要がある。たとえ安価なメモリを使用しても、この様なミラーリングに関連する追加コストは複数のサイトにミラー化するには膨大となり、これはほとんどの使用者が未だミラー・サイトから遠く離れていることを意味する。ミラーリングはまたコンテンツ・プロバイダに対して、全てのサイトが整合して最新であることを保証しなければならないという別の欠点を有し、これはたとえ数サイトであっても軽視出来ない仕事である。20

【0079】

本発明では、コンテンツは自動的にグローバル・サーバ・ネットワークに、インテリジェントかつ効率的なやり方で複写される。コンテンツはそれらが必要とされる場所にのみ複写される。更に、コンテンツが変更されると、新たなコピーが好適にネットワークを通して自動的に複写される。30

【0080】

3. 自動的コンテンツ管理：

多くの既存の解法は、コンテンツ配信、コンテンツ複写および異なるサーバ間での負荷バランスに対して能動的な管理を必要とする。特に、コンテンツを何処で担当するかの決定は手動で行われなければならず、またデータを複写する処理工程は、中央化されたプッシュ方式で処理される。これに対して、本発明の特徴は受動管理である。複写は要求ベースのプル方式で、コンテンツが本当にそれを必要としている所にのみ送られるように、好適になされる。更に、処理工程は好適に全自動化されている；ISPはコンテンツが如何にしてまた何処に複写されるべきかを悩む必要は無い、そして／またはコンテンツ・プロバイダも同様である。40

【0081】

4. 無制限な、経済的拡張性：

競合する解法は拡張性が低く、数個のサイトまでである。例えば、ミラーリングに基づく解法は典型的に最大3または4サイトを接続して使用される。規模拡大の障壁には、全サイトを複写する費用、全てのノードでの計算資源を複写するコスト、およびコンテンツ・プロバイダが彼らのサーバ上で使用する多数の種類のソフトウェア・パッケージを管理する複雑さが含まれる。

【0082】

本発明のユニークなシステム・アーキテクチャは数百、数千または百万個のノードまで拡50

張可能である。ホスティング・ネットワーク内のサーバは異常状態になったり故障する可能性があるが、システムの全体機能は影響されない。グローバル・ホスティング枠組み構造は資源の有効利用を行う；サーバおよびクライアント・ソフトウェアは全てのノードで複写される必要は無いが、それはホスティング・サーバのみが各ノードで動作しているからである。加えて、グローバル・ホスティング・サーバは高い信頼性を要求されない、標準の簡単なハードウェア上で動作するように設計されている。

【0083】

5. 突然の負荷集中に対する防護：

競合する解法はコンテンツ・プロバイダに対して、予期していない突然の負荷集中に対する防護策を提供していない。ミラーリングおよび関連する負荷バランス解法は、コンテンツ・プロバイダが負荷をサーバの集合の中で分配することを可能とするが、サーバの集合容量はピーク時要求を処理するのに十分でなければならない。これはコンテンツ・プロバイダが実際の平均負荷の代わりに予想されるピーク負荷に見合ったレベルの資源を購入し保守しなければならないことを意味する。インターネットの高度に変化が激しくまたは予想できない性質を考えると、この様な解法は高価で資源の無駄遣いである。

10

【0084】

本発明のホスティング・アーキテクチャは、ISPがコンテンツ・プロバイダに対して突然の負荷集中保証を提供するためにホスティング・サーバの単一ネットワークを利用するこ²⁰とを可能とする。すなわち、プロバイダ・サイト上での予期せぬ高い負荷に対して自動的に適合しサポートする保証である。ISPは多くのプロバイダを同一グローバル・ネットワーク上で集合しているので、資源は更に効率的に使用される。

【0085】

6. 実質的帯域幅節約：

競合する解法は、ISPまたはコンテンツ・プロバイダに対して実質的な帯域幅節約は提供しない。ミラーリングの使用を通して、あるリンク（すなわち、ニューヨークとロスアンゼルス間）に関して帯域幅を節約することが可能である。しかしながら、グローバル・ホスティング無しでは、コンテンツに対するほとんどの要求はインターネットを通過する必要があり、従って帯域幅コストを増加させる。本発明のホスティング枠組み構造は、それら自身の基幹を有するISPに対する事実上の基幹帯域幅を節約する。コンテンツはネットワークを通して配信され、ネットワーク交換ポイントの隣に配置することが出来るので、ISPおよびコンテンツ・プロバイダは共に基幹料金がほとんどのコンテンツ要求に課せられないため、事実上の節約を経験する。

30

【0086】

7. グローバル・ネットワークへの瞬時アクセス：

競合する解法はコンテンツ・プロバイダに対して、コンテンツを担当、そして／または複写するサイトの小さな集合を手動で選択するように要求する。たとえISPが広範に広がっている場所に多数のホスティング・サイトを有していたとしても、特別に選択された（および対価が支払われている）これらのサイトのみが、そのコンテンツ・プロバイダに対するコンテンツを担当するために使用される。

40

【0087】

これに対して、本発明のグローバル・ホスティング解法は、ISPが彼らのクライアントに対してサーバのグローバル・ネットワークへの瞬時のアクセスを提供することを可能とする。グローバル・ネットワークへの瞬時のアクセスを提供するために、コンテンツは好適に絶え間なくまたダイナミックにネットワーク内を移動される。例えば、コンテンツ・プロバイダがアジア地区の顧客に興味のあるコンテンツを追加した場合、コンテンツ・プロバイダはそのコンテンツがアジア地区に配置されているサーバに自動的に移動されることを保証される。加えて、グローバル・ホスティング枠組み構造は、コンテンツが最終使用者の非常に近くに（企業市場の場合使用者の建物の様な近くに）移動されることを可能とする。

【0088】

50

8. グローバル I S P および集成体への設計 :

ほとんどの競合する解法はコンテンツ・プロバイダで購入され管理されるように設計されており、彼らの多くは既に単一サーバを管理ための統括管理および運用作業に対して絶え間なく挑戦し消耗している。本発明のホスティング手法はグローバル I S P で配備され、コンテンツ・プロバイダに提供出来る新たなサービスを提供する。このサービスの特徴は、コンテンツ・プロバイダに要求される運営および管理を最小とし、従ってコンテンツ・プロバイダがユニークなコンテンツを制作するというその中枢ビジネスに集中出来るようになる。

【 0 0 8 9 】

9. 固有データベースおよび秘密情報の効果的管理 :

多くの競合する解法は、コンテンツ・プロバイダに対して彼ら固有のデータベースを地理的に遠く離れた複数のサイトへ複写することを要求する。その結果、コンテンツ・プロバイダはその固有のまた通常秘密であるデータベースに対する効果的な管理を失ってしまう。これらの問題を取り除くために、本発明のグローバル・ホスティング解法はコンテンツ・プロバイダが彼らのデータベースに対する完全な管理を維持することを保証する。先に説明したように、コンテンツに対する最初の要求はコンテンツ・プロバイダの中央ウェブ・サイトに向けられ、これは統一して効果的かつ管理されたデータベース・アクセスを実施する。好適に、ページ要求に対する高帯域幅、固定部分はグローバル・ホスティング・ネットワークからリトリーブされる。

【 0 0 9 0 】

10. コンテンツ・プロバイダのソフトウェアとの互換性 :

多くの競合する解法はコンテンツ・プロバイダに対して特定セットのサーバとデータベースを使用することを要求する。これらの特に、一定しない要求はコンテンツ・プロバイダが最も効率的に最新技術を使用する能力に制約を与え、コンテンツ・プロバイダの既存基盤構造をお金を掛けて変更する必要を迫る場合もある。これらの問題を取り除くことにより、本発明のグローバル・ホスティング・アーキテクチャは効果的にコンテンツ・プロバイダと I S P との間のインターフェースを行い、これはコンテンツ・プロバイダで使用されているシステムまたはサーバに対して何らの前提も必要としない。更に、コンテンツ・プロバイダのシステムは本発明のアーキテクチャを修正変更したりまたは中断することなく、更新、変更または完全な置き換えを行うことが可能である。

【 0 0 9 1 】

11. ダイナミック・コンテンツ、個人化された広告または E コマース、と陳腐化されていないコンテンツとの間の相互干渉の排除 :

多くの競合する解法（例えば、全てのコンテンツの単純なキャッシング）はダイナミック・コンテンツ、個人化された広告および E コマースと相互干渉し、使用者に対して陳腐なコンテンツをサービスする可能性がある。その他のソフトウェア会社はこれらの問題を部分的に取り除く試みを続けているが（例えば、全てのキャッシングされたコピーに対するヒット計数の保存）、これらの解法の各々は機能の部分的または完全に失わせかねない（例えば、広告を個人化させる機能）。これに対して、グローバル・ホスティング解法はダイナミック・コンテンツ、個人化された広告または E コマースと相互干渉することは無く、それはこれらの作業の各々が好適にコンテンツ・プロバイダの中央サーバで処理されるためである。

【 0 0 9 2 】

12. グローバル・ネットワーク対応設計 :

グローバル・ホスティング・アーキテクチャは高度な拡張性を有し、従って全世界規模ネットワーク基盤上に配備できる。

【 0 0 9 3 】

グローバル・ホスティング・アーキテクチャの各々の構成要素に関して先に説明した機能は、好適にプロセッサの中で実行可能なソフトウェアで実現される、すなわち、コンピュータの隨意アクセスメモリ内に存在するコード・モジュール内の命令またはプログラム・

10

20

30

40

50

コードの組で実現される。コンピュータから要求されるまで、命令の組は別のコンピュータ・メモリ、例えばハードディスク・ドライブの中、または光磁気ディスク（C D R O M内での使用される場合）またはフロッピー・ディスク（フロッピー・ディスク・ドライブ内で使用される場合）の様な携帯型メモリの中に格納されており、またはインターネットまたはその他のコンピュータ・ネットワークを経由してダウンロードされる。

【0094】

加えて、説明された種々の方法は汎用コンピュータの中で選択的に起動したりソフトウェアで再構成することで簡便に実現できるが、通常の技術を有する当業者は、その様な方法がハードウェア、ファームウェア、または更に要求された方法のステップを実行するよう構築された特殊装置で実現できることを理解するであろう。

10

【0095】

更に、ここで使用されるように、ウェブ「クライアント」は広く解釈されるべきであり、インターネットのような、全ての既知のまたは今後開発されるであろうコンピュータ・ネットワークに直接または間接的に接続されていたりまたは接続可能な、全てのコンピュータまたは構成要素を意味する。用語ウェブ「サーバ」は広く解釈されるべきであり、コンピュータ、コンピュータ・プラットフォーム、コンピュータまたはプラットフォームの付属物、またはそれらの全ての構成要素を意味する。もちろん「クライアント」は広く解釈されるべきであり、ファイルを要求または獲得する人を意味し、また「サーバ」はファイルをダウンロードするエンティティーである。

【0096】

20

此処まで我々の発明を説明してきたが、我々が新たに請求し要望する特許内容は添付の特許請求の項に記載されている。

【図面の簡単な説明】

【図1】 図1はその中で本発明が実現されている代表的なシステムである。

【図2】 図2は基本ドキュメントおよび組み込みオブジェクトの組を図示するマークアップ言語ドキュメントの簡略化された表現である。

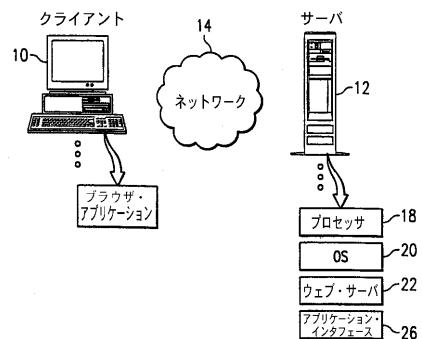
【図3】 図3は本発明に基づく全世界ホスティング・システムの高レベル図表である。

【図4】 図4はウェブ・ページを本発明で使用される修正された組み込みオブジェクトU R Lに処理するための方法を図示する簡略化された流れ図である。

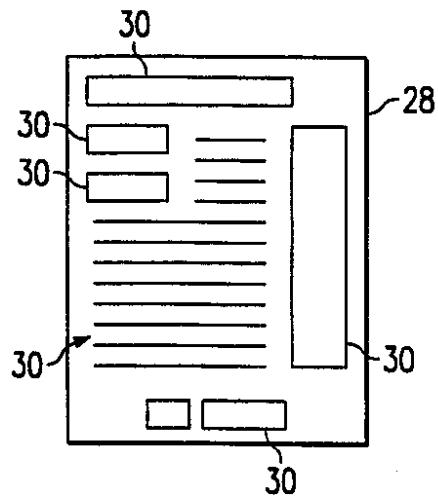
【図5】 図5はウェブ・ページに対するH T T P要求に、本発明がいかに応答するかを図示する簡略化された状態図である。

30

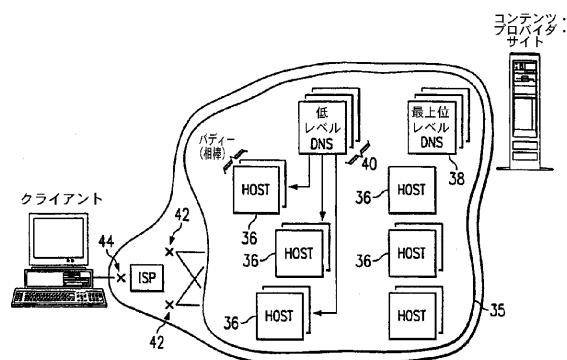
【図1】



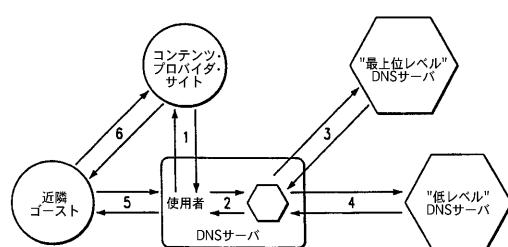
【図2】



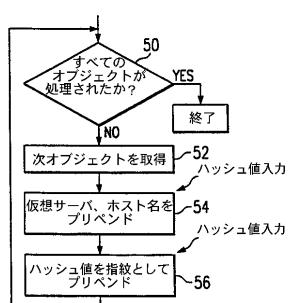
【図3】



【図5】



【図4】



フロントページの続き

(72)発明者 レイトン、エフ、トムソン

アメリカ合衆国 マサチューセッツ、ニュートンビル、チャールズデン パーク 15

(72)発明者 ルイン、ダニエル、エム

アメリカ合衆国 マサチューセッツ、ケンブリッジ、バッサー ストリート 292、ナンバー
ディ6

合議体

審判長 小曳 満昭

審判官 池田 聰史

審判官 長島 孝志

(56)参考文献 國際公開第97/29423 (WO, A1)

特開平8-44643 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F12/00

G06F13/00