

ユビキタス環境に向けた分散コンテンツ発見

中内 清秀[†] 森川 博之[†] 青山 友紀[†]

[†] 東京大学工学系研究科 〒113-8656 東京都文京区本郷 7-3-1
E-mail: †{nakauchi,mori,aoyama}@mlab.t.u-tokyo.ac.jp

あらまし ネットワーク上に膨大な数の共有コンテンツ及びその複製やキャッシュが遍在するユビキタスコンテンツ環境においては、ユーザがいつでもどこでも所望のコンテンツを確実にかつ効率的に発見し、そのコンテンツをユーザの置かれた環境に適した形で容易に利用できることが期待される。本稿では、まずこのようなコンテンツの発見及び配信をサポートする広域コンテンツ共有インフラ、ユビキタスコンテンツネットワーク (UCN) の基本概念を示す。次に、UCN の重要機能である透過的コンテンツアクセス機能に関して、関連性のある複数の共有コンテンツに対する同時アクセスを高速化することを目的とした、コンテンツの関連性を用いた効率的な P2P 分散コンテンツ発見機構を示す。

キーワード 分散コンテンツ発見, ピアツーピアネットワーク, ユビキタスネットワーク

Distributed Content Location for Ubiquitous Environments

Kiyohide NAKAUCHI[†], Hiroyuki MORIKAWA[†], and Tomonori AOYAMA[†]

[†] School of Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan
E-mail: †{nakauchi,mori,aoyama}@mlab.t.u-tokyo.ac.jp

Abstract In ubiquitous content environments where enormous shared contents could be replicated or cached everywhere in the networks, the mechanisms are essential for users to effectively discover desired contents and to receive transformed contents suitable for their environments. In this paper, we first describe basic concepts of Ubiquitous Content Network (UCN), a global-scale content sharing infrastructure which provides such mechanisms. Then, we show an effective Peer-to-Peer (P2P) distributed content location mechanism using content-relationship for the purpose of fast transparent access to multiple related contents.

Key words Distributed Content Location, Peer-to-Peer Network, Ubiquitous Network

1. はじめに

端末やデバイスの高性能化, FTTH, ADSL, 無線 LAN, Bluetooth などによるインターネット常時接続環境の整備という背景のもと, PC のみならず, 携帯電話, PDA, ビデオカメラ, センサなどのあらゆるデバイスがインターネットに接続され, 互いに通信することにより人の社会生活をサポートするユビキタス環境 [1] が実現されつつある。一方, インターネットアクセスの広帯域化, ストレージデバイスの低廉化などにより, Gnutella [2] や Freenet [3] に代表される P2P 型コンテンツ流通モデルが出現し, WWW や CDN を含めてコンテンツ流通形態の多様化が進んでいる。P2P 型コンテンツ流通モデルでは, 従来単なる受動的なクライアントであった PC などのノードが, オーバレイネットワークを形成しつつ能動的で強力なサーバとして機能し, ノード同士が直接コンテンツを送受信する。

このようなコンテンツ流通形態の多様化に伴い, ネットワークを構成するノード数だけでなく, 今後ますます共有コンテンツの数が膨大になることが予想される。例えば, 1 人 1 テラバイト (10^{12} Byte) のコンテンツを P2P コンテンツ共有システム上で公開した場合, 地球上におよそ 5 ゼタバイト (5×10^{21} Byte) の巨大な超分散データベースが構築されることになる。この時, 多数のユーザにより共有されるコンテンツについては, コンテンツの流

通が進むにつれて多数の複製が生成されることになる。同時に, 信頼性, 可用性, 負荷分散, ネットワーク利用効率という観点からも, その複製 (コピー, レプリカ) やキャッシュが複数生成されるであろう [4]。このように, ユビキタス環境において共有コンテンツ数が爆発的に増加すれば, インターネット上に共有コンテンツ及びその複製やキャッシュ, 即ち多数の同一コンテンツが遍在する「ユビキタスコンテンツ環境」(図 1) が実現されると考えられる。

本研究の根本的な目標は, ユビキタスコンテンツ環境において, ユーザがいつでもどこでも所望のコンテンツを確実にかつ効率的に発見し, そのコンテンツをユーザの置かれた環境 (コンテキスト) に適した形で容易に利用することを可能とするグローバルスケールのコンテンツ共有インフラの構築である。このようなインフラを「ユビキタスコンテンツネットワーク (UCN)」と呼ぶ。UCN はユーザ同士のコンテンツ共有 (C2C) だけでなく, ユーザとコンテンツプロバイダ (B2C), コンテンツプロバイダ同士 (B2B) のコンテンツ共有をも柔軟にサポートすることを目指す。

UCN は 1) コンテンツ発見機構と 2) コンテンツ配信機構から構成される。コンテンツ発見機構の基本機能は, ユーザが一意指定した共有コンテンツへの位置透過的かつ複製透過的なアクセス (透過的コンテンツアクセス) を実現するコンテンツ発見や, Gnutella のようなユーザが指定したメタデータ (キーワード) を

もつ共有コンテンツの検索である。この時、効率的な透過的コンテンツアクセスやコンテンツ検索を実現するためのコンテンツネーミング機能も必要となる。一方、コンテンツ配信機構の基本機能は、マルチキャスト、モビリティのサポート、QoS 保証などである。UCN はこれらの基本機能を P2P による完全分散の手法を用いて実現する。

本稿では、上述の UCN の基本機能のうち最重要機能である、透過的コンテンツアクセスを実現する分散コンテンツ発見機能について述べる。ここで透過的とは、ユーザがネットワーク上のどこに存在しても（位置透過性）、また複数の同一コンテンツがネットワーク上に存在しても（複製透過性）、複数の同一コンテンツの中から常に最適な共有コンテンツを効率的に発見することである。この時、ユーザは発見対象となるコンテンツをコンテンツに付与された GUID（Globally Unique Identifier）を用いて一意に指定できるものとする。

一般的に GUID により一意に指定されたリソースやサービスをインターネットのような広域ネットワーク上で発見する手法としては、DNS や CDN [5]（CDN 対応 DNS）などが普及しているが、これらの手法は階層構造や CDN 事業者による CDN 対応 DNS の集中管理という仕組みを取り入れていることから、ユビキタスコンテンツ環境のような現在よりもはるかに膨大な数のコンテンツの発見や、GUID として URL のような階層型ネーミング機構を採用しないコンテンツの発見などにもそのまま適用できるとは考えがたい。

筆者らは、ユビキタスコンテンツ環境では、膨大な数の共有コンテンツに対しても効率的な透過的コンテンツアクセスを実現できるポテンシャルをもつ「P2P 分散コンテンツ発見」に注目している。P2P 分散コンテンツ発見手法では、クエリルーティングと指定された GUID の名前解決を統合的に行う分散ハッシュテーブル（DHT: Distributed Hash Table）[6]～[11]を用いることにより透過的なコンテンツアクセスを可能としている。DHT を用いてコンテンツ発見を行う場合、一般的にクエリーのホップ数は $O(\log N)$ （ N ：ノード数）であることが知られており、効率的なコンテンツ発見によりスケラビリティを確保できる。

しかしながら、従来の P2P 分散コンテンツ発見手法では、例えば同一ディレクトリ内のコンテンツ群や一つのサービスを構成するコンテンツ群などのように、関連性が強いと考えられる複数の共有コンテンツを同時に発見する場合でも、個々のコンテンツに対して独立に DHT を用いて発見を行っている。そのため、同時にアクセスする関連コンテンツの数が増加すれば、それに比例してクエリーのホップ数の総和が増加し、ネットワークの負荷増大を招いてしまう。

そこで本稿では、コンテンツの関連性に着目し、関連性のあるコンテンツ群に対して同時アクセスを行う場合に、関連性を利用した効率的なコンテンツ発見手法 C-DHT を示す。具体的には、関連性のあるコンテンツ群をクラスタリングし、クラスタの発見には広域でも効率的な DHT を、クラスタ内での個々のコンテンツの発見には高速な局所的コンテンツ発見手法をそれぞれ利用することにより、全体としてクエリーのホップ数の総和を抑制する。

本稿の構成を以下に示す。まず、2. において UCN の概要を示す。次に、3. において一般的な P2P 分散コンテンツ発見手法について述べ、4. においてコンテンツの関連性を利用した P2P 分散コンテンツ発見手法を示す。最後に、5. においてまとめを述べる。

2. ユビキタスコンテンツネットワーク（UCN）

2.1 UCN の概要

UCN はユーザがいつでもどこでも所望のコンテンツを確実に

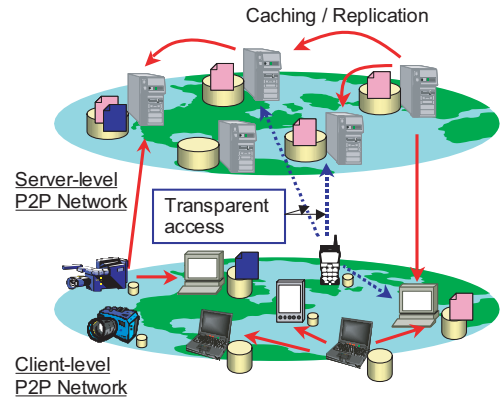


図 1 ユビキタスコンテンツ環境
Fig. 1 ubiquitous Content Environments

つ効率的に発見し、そのコンテンツをユーザの置かれた環境（コンテキスト）に適した形で容易に利用することを可能とするグローバルスケールのコンテンツ共有インフラである。UCN はコンテンツ発見機構とコンテンツ配信機構から構成され、これらの機能を統合的に提供する。P2P による完全分散システムが、ユビキタスコンテンツ環境のような膨大な数の共有コンテンツに対する効率的な透過的アクセスや検索を実現できるポテンシャルをもつことから、UCN ではこれらの機構を P2P で実現する。これにより、ユーザは特定のサーバに管理されることなく、自由なコンテンツ発見、コンテンツ共有、コンテンツ交換を行うことができる。

UCN においては、ユーザが指定した共有コンテンツ、またはユーザの希望する条件を満たす共有コンテンツを効率よく検索、発見する「コンテンツ発見」、マルチキャスト配信や端末のモビリティサポート、QoS 保証を行う「コンテンツ配信」の 2 つが重要な機能となる。2.4 及び 2.5 において、求められる基本機能について述べる。

2.2 サービスモデル

UCN が想定するサービスモデルは、図 1 に示すように、主にサーバレベル P2P モデル、及びクライアントレベル P2P モデルに分類することができる。

サーバレベル P2P モデルでは、ネットワーク内に分散配置されたサーバを構成要素とする P2P ネットワークが構築される。一般にサーバはコンテンツ事業者やコンテンツ配信事業者により管理されるため、信頼性の高いサービスが実現される。クライアントは、P2P ネットワークのエントリーポイントとなるサーバに接続することにより、サービスを楽しむことができる。コンテンツ事業者によるコンテンツ複製管理（B2B）や P2P 型 CDN（B2C）などの適用分野が考えられる。

一方、クライアントレベル P2P モデルでは、クライアントのみを構成要素とする P2P ネットワークが構築される。サービスを実現するためのインフラ（サーバ）が不要であるため、信頼性が求められないようなサービスを迅速に展開することができる。クライアント同士のコンテンツ共有やアドホックな小規模テレビ会議（C2C）などの適用分野が考えられる。

上記のどちらのモデルを採用しても機能的に同一のサービスを実現できると考えられるが、実際には性能、信頼性、コスト、匿名性、セキュリティなどの要求レベルを総合的に考慮した上で、これらのモデルが使い分けられると考えられる。本稿で述べる分散コンテンツ発見機構では、主にサーバレベル P2P モデルを想定するが、両モデルの優劣についての検討は今後の課題とする。

2.3 デザイン指針

UCN のコンテンツ発見、配信機構の設計には、以下のような

点が要求される [4], [12] .

スケーラビリティ：コンテンツ数，ノード数の増加に対して，コンテンツ発見，配信のパフォーマンスの低下を抑制できないといけない．また，性能，ストレージ容量に応じてコンテンツ管理やコンテンツ配信の負荷を配分するような負荷分散を行わなければならない．

ロバスト性：特定のノードやネットワークの障害に対して，リソース発見機構全体の機能の停止を防止できないといけない．また，ノードが保持するリソース情報について，ノード間で一貫性を完全に維持できなくても，サービスが継続できなければならない．同時に，メンテナンス負荷が最小になるよう，自律分散性も必要となる．

応答性：目的のコンテンツがネットワーク上に存在する場合，必ずそのコンテンツを発見できなければならない．

位置非依存性/ローカルリティ：ネットワーク内にコンテンツの複製やキャッシュが複数存在する状況においても，かつ，ノードがネットワークのどこに接続されていても，常に最適なコンテンツを透過的に発見できることが望まれる．

2.4 コンテンツ発見機構

コンテンツ発見機構は，透過的コンテンツアクセスを実現するための分散コンテンツ発見機能，分散コンテンツ検索機能，コンテンツネーミング機能から構成される．

分散コンテンツ発見（透過的コンテンツアクセス）

名前や GUID (Globally Unique Identifier) が既知であるコンテンツにアクセスするために，そのコンテンツを保持しているノードの IP アドレスを獲得するコンテンツ発見 (content location) を行う．具体的には，コンテンツの複製やキャッシュが多数存在する場合，高速アクセス，負荷分散などの観点から，複数の同一コンテンツの中からホップ数やサーバ負荷などのメトリックに従って最適なコンテンツを発見する．位置非依存なコンテンツネーミングに基づく位置非依存クエリルーティングが課題となる [6] ~ [11] . 本稿では，この分散コンテンツ発見機能について述べる．

分散コンテンツ検索

コンテンツに属性やキーワード (メタデータ) が付与されている環境においては，指定されたメタデータをもつ (条件を満たす) コンテンツの一覧及びそのコンテンツを保持するノードの IP アドレスを獲得する．膨大な数のコンテンツに対するスケーラビリティの観点から，検索エンジンのような集中型の検索ではなく P2P ネットワークを利用した分散型の検索を行う．Gnutella のようなブロードキャストによるクエリルーティングではスケーラビリティが問題となるため，効率的なクエリルーティングが課題となる [3], [13] .

コンテンツネーミング

ユビキタスコンテンツネットワークにおいて「最寄りのコンテンツの発見」というような位置非依存なコンテンツアクセスを実現するためには，WWW における URL のような位置を示した位置依存コンテンツネーミングではなく，コンテンツそのものを表す位置非依存コンテンツネーミングが必要である．例えばユニークなファイル名，コンテンツ内容のハッシュ値などが考えられる．このような位置非依存なコンテンツネーミングにおいても，コンテンツの集合体に対するある程度汎用的な名前集約 (抽象化) が求められる [12], [14] .

2.5 コンテンツ配信機構

コンテンツ配信機構は，アプリケーションレベルマルチキャスト機能，モビリティサポート機能，QoS 保証機能から構成される．

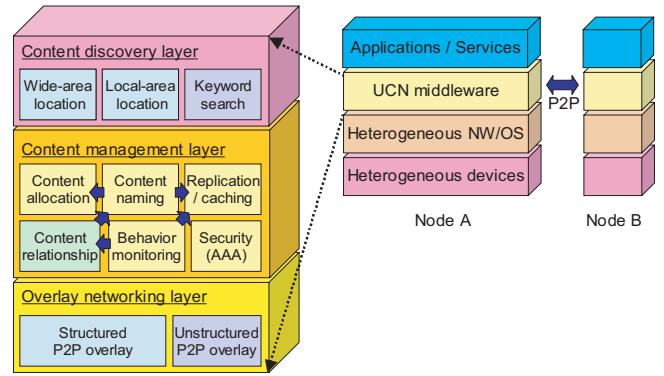


図 2 分散コンテンツ発見のための UCN アーキテクチャ
Fig.2 UCN Architecture for Distributed Content Discovery

アプリケーションレベルマルチキャスト

特定のノードが保持する 1 つのコンテンツに対して同時にアクセス要求が到着した場合には，マルチキャスト機能が有効である．しかしながら，IP 層で実現する IP マルチキャストが本質的にもつ多数の難題 [15] のため，そのインフラ整備は進んでいない．そこで，マルチキャスト機能をすべてアプリケーション層で実現するアプリケーションレベルマルチキャスト (ALM) が必要となる [16] . ALM では，参加ホストが構築したオーバレイネットワーク上で，ホスト間のユニキャストを連続的に利用することにより，最終的に全参加ホストへのデータ配信を実現する．筆者らは現在，ALM を提供するための汎用的なミドルウェアに関して検討を行っている [17] .

モビリティサポート

モバイルホストに対して継続的にコンテンツを配信する場合，ハンドオフによるコンテンツ配信の中断を防止しなければならない．そのために，アプリケーションに対してターミナルモビリティを隠蔽し，ネットワーク間の移動に対してシームレスに同一のコンテンツを継続的に受信できるターミナルモビリティサポート機構が必要である．同時に，移動中でも常に最適なコンテンツサーバから同一コンテンツを継続的に受信するために，動的サーバ切替を行うコンテンツモビリティサポート機構 (モバイル CDN) が必要である．ただし，最適なサーバの発見は既存の CDN のような集中型ではなく，P2P による分散型で行う．

QoS 保証

ユーザに対してコンテンツ配信の品質を保つためには，ユーザのネットワーク接続状態，コンテンツアクセスに利用される機器の種別及びその性能，希望されるコンテンツ表現形式などのユーザのコンテキスト情報を抽出し，コンテンツをコンテキストに適したフォーマットやレートでユーザに配信する必要がある．そのためには，コンテンツをネットワーク内に配置されたトランスコードにより変換する機構やレート制御機構が必要である．これに向けて，ユーザのコンテキストを抽出し管理するコンテキスト抽出管理機構，トランスコード発見機能が必要である．また，ネットワークの輻輳状態を検出し，輻輳箇所を経由しないコンテンツサーバを発見する機構も必要である．

2.6 UCN アーキテクチャ

本稿では UCN における分散コンテンツ発見機能に焦点を当てているため，2.4 で示したコンテンツ発見機構に関する機能だけを抽出した UCN アーキテクチャを図 2 に示す．

まず，UCN の階層構造を説明する．UCN は，各ノードにおいてネットワーク層の上位の汎用的ミドルウェア (UCN ミドルウェア) として実現される．各ノードの UCN ミドルウェア同士の協調により P2P ネットワークが構築される．UCN ミドルウェアは，

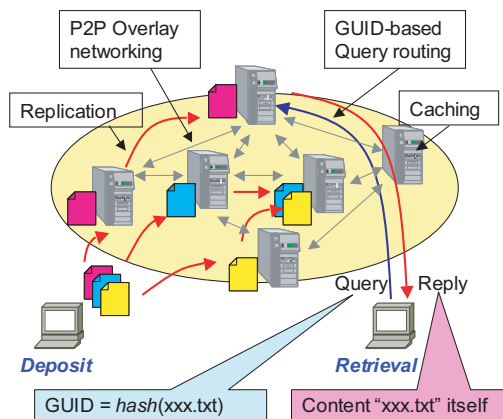


図3 P2P 分散ストレージ
Fig. 3 P2P Distributed Storage

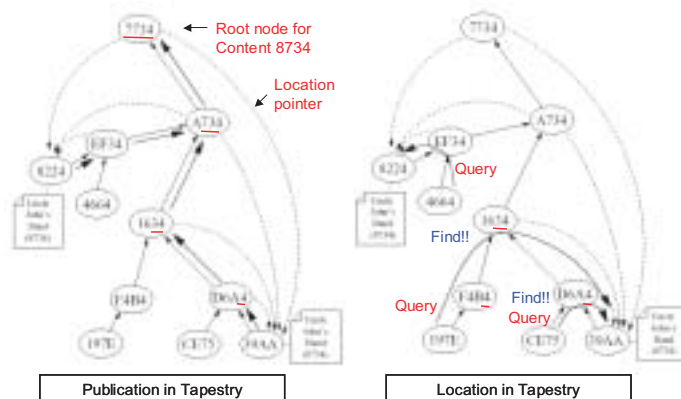


図4 Tapestry
Fig. 4 Tapestry

上位に位置するアプリケーションからコンテンツアクセス要求を受信した場合、適当なコンテンツを発見するとともに、配信されてきたコンテンツそのもの、またはコンテンツの位置情報（IP アドレス）をアプリケーションに返す。

次に、UCN ミドルウェアの構成を簡単に説明する。UCN ミドルウェアはオーバーレイネットワーク層、コンテンツ管理層、コンテンツ発見層から構成される。オーバーレイネットワーク層は P2P ネットワークのトポロジーの維持、最適化を行う。コンテンツ管理層はコンテンツの複製やキャッシュの作成/削除、その配置場所の管理、セキュリティ管理などを行う。4. で述べるコンテンツの関連性の管理などもこの層で行う。最後に、コンテンツ発見層は P2P ネットワーク上でクエリルーティングを行い、2.4 で示した分散コンテンツ発見や分散コンテンツ検索を行う。

3. P2P 分散コンテンツ発見

本節では、本稿で想定するサービスモデルである P2P 分散ストレージ、及び P2P 分散ストレージを実現するための要素技術について述べる。

3.1 P2P 分散ストレージ

P2P 分散ストレージ [18], [19] とは、個々のホストのローカルストレージ（の一部）がネットワーク経由で P2P で直接接続、共有されることにより構築される 1 つの巨大な仮想ストレージである。P2P 分散ストレージでは、コンテンツの GUID を入力とした時に、そのコンテンツそのものを出力とする広域コンテンツ発見機構である。P2P 分散ストレージの概要を図 3 に示す。

分散ストレージはファイルシステムの下層に位置し、ファイルシステムの上に位置するアプリケーションは、どのノード内のストレージかを意識することなく、ローカルストレージにアクセスするかのように他ノード内のストレージにアクセスすることができる。つまり、あるホストが所有者であるコンテンツは、他のホストのローカルストレージに自由に保存される。

また、負荷分散、高速アクセス、ネットワーク帯域の利用効率の向上を目的として、コンテンツは必要に応じて複製やキャッシュが生成される。複製やキャッシュも同様にすべてのホストのローカルストレージに自由に保存することができる。

更新が可能なコンテンツについては、複数のノード内に存在する複製間で一貫性を保持するために同期を行う。ただし、同期メカニズムについては分散コンテンツ発見機構と独立に論じられるため、本稿では触れない。本稿ではマルチメディアコンテンツのような一般的に更新頻度の少ないコンテンツを対象とする。

3.2 要素技術

P2P 型分散ストレージの要素技術として、分散ハッシュテーブル (DHT: Distributed Hash Table) [6] ~ [11] と Attenuated Bloom Filters [4], [20] を説明する。前者は広域コンテンツ発見に、後者は局所的コンテンツ発見に適している。

3.2.1 DHT

DHT ではすべてのコンテンツ及びノードにそれぞれユニークな GUID、ノード ID (NodeID) が付与される。GUID はコンテンツ名のハッシュ値やコンテンツ内容のハッシュ値などが用いられる。NodeID も同様に、IP アドレスのハッシュ値や、ランダムに選択された値などが用いられる。ハッシュ関数としては、ハッシュ値の桁数を増加させても、すべての GUID、NodeID を求め直す必要のない一貫性ハッシュ関数 [21] を用いる。Chord [8] では、GUID、NodeID とともに 160 bit で表されている。

DHT では、NodeID に基づいて構造的なオーバーレイが構築され、コンテンツ（またはその位置情報、ポインタ）は GUID と NodeID が一致するノードにより管理される。このオーバーレイ上で位置非依存な GUID を用いてクエリルーティングを行うことにより、位置透過的、複製透過的コンテンツ発見を可能とする。コンテンツの発見に必要なクエリの（オーバーレイ上の）ホップ数は、最大 $O(\log N)$ であることが証明されている (N : ノード数) [11]。

Tapestry [6], [7] を例に、DHT の動作例を説明する。Tapestry は分散ストレージシステム OceanStore [4] のための DHT である。Tapestry では、ホストは保持するコンテンツのポインタ（コンテンツ名と自身の IP アドレスの組）を分散ストレージに登録する。

Tapestry におけるオーバーレイは、NodeID のサフィックスをもとに構築される。例えば、NodeID 7734 のノードは、GUID が下の桁から順に i 桁 ($0 \leq i < 3$) だけ一致するノード (NodeID = 51E5, 9C34, 2A34, A734) を隣接ノードとして論理リンクを張る。

Tapestry におけるコンテンツの登録の様子を図 4 の左に示す。NodeID 39AA のノードは GUID 8734 のコンテンツを登録するために、(GUID = 8734, NodeID = 39AA) というコンテンツの位置情報（ポインタ）を示したクエリを、コンテンツのルートノード (NodeID 8734) に向けて、隣接ノード D6A4 に送信する。このクエリは、GUID の一致する下の桁数が 1 つずつ増加するように、即ち、D6A4 1634 A734 7734 という NodeID をもつノードを転送されていく。^(注 1) つまり、サフィックスペー

(注 1) : NodeID 8734 をもつノードが存在しない場合には、GUID 8734 に一致する下の桁数が最も多い NodeID 7734 をもつノードがルートノードとなる。

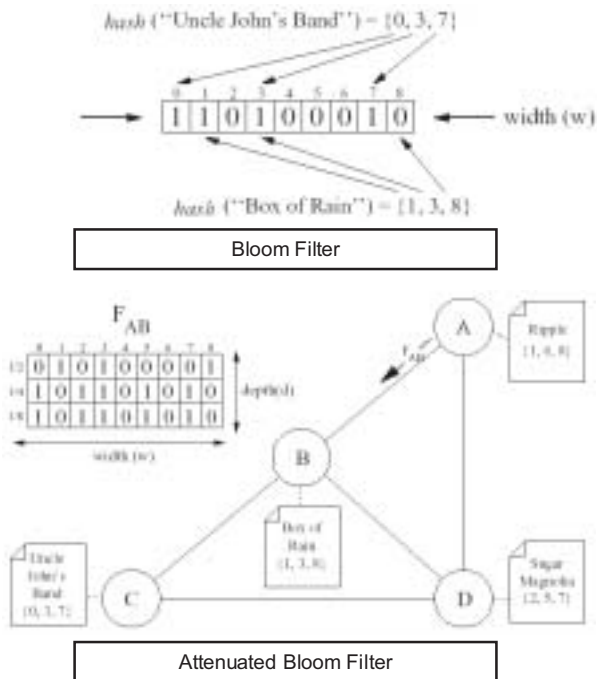


図 5 Attenuated Bloom Filters
Fig. 5 Attenuated Bloom Filters

スのルーティングを行う。この経路上のノードがこのコンテンツの位置情報を保持する。

Tapestry におけるコンテンツ発見の様子を図 4 の右に示す。NodeID 197E のノードが GUID 8734 のコンテンツを発見する場合、GUID 8734 を指定したクエリーをコンテンツのルートノード (NodeID 8734) に向けて、隣接ノード F4B4 に送信する。クエリーは、F4B4 1634 と転送された時点で、ノード 1634 が保持する (GUID = 8734, NodeID = 39AA) を発見できるため、クエリーはノード 39AA にリダイレクトされる。このようにしてコンテンツ 8734 が発見される。

3.2.2 Attenuated Bloom Filters

DHT は広域コンテンツ発見には有効であるが、クエリーを出すホストから目的のコンテンツまでのホップ数が小さい場合には有効ではない [20]。そこで、局所的な P2P 分散コンテンツ発見手法として Attenuated Bloom Filters [4], [20] が提案されている。Tapestry と同様、ホストは保持するコンテンツのポインタ (コンテンツ名と自身の IP アドレスの組) を分散ストレージに登録する。

Attenuated Bloom Filters の動作例を図 5 を用いて説明する。例えば、図 5 の上の図において、保持するコンテンツ "Box of Rain" をもつホストは、 n 個 (ここでは $n = 3$) の異なる共通のハッシュ関数を用いて、そのハッシュ値 (ここでは $\{1, 3, 8\}$) を w bit (ここでは $w = 8$) の Bloom Filter に格納する。他の保持するコンテンツについても同様の処理を行い、1 つのノード内のすべてのコンテンツの情報を 1 つの Bloom Filter に集約させる。

次に、図 5 の下の図のように、各ホストの Bloom Filter を P2P ネットワークの隣接ホスト間で交換して伝播させることにより、アウトプットインタフェース毎に深さ d (ここでは $d = 3$) の Bloom Filter の配列 (Attenuated Bloom Filters) を作成する。この配列がルーティングテーブルに相当し、 i 列目が i ホップ先のコンテンツを表す。例えば、ホスト A のホスト B へのリンクの配列 F_{AB} において、1 列目の Bloom Filter では 1 ホップ先のホスト B のもつコンテンツのハッシュ値 ($\{1, 3, 8\}$) のビットが立ち、2

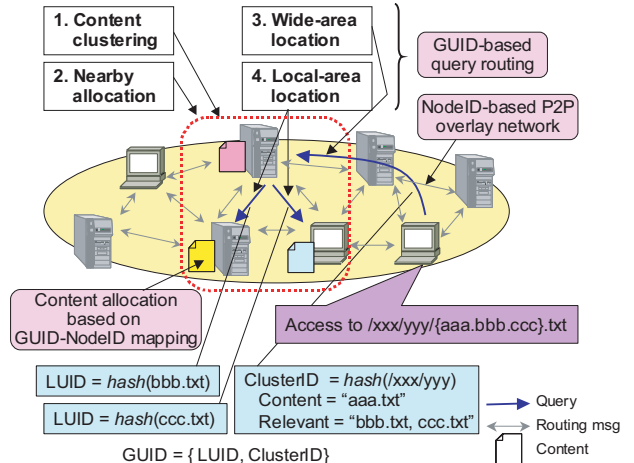


図 6 C-DHT: コンテンツの関連性を用いた分散コンテンツ発見
Fig. 6 C-DHT: Distributed Content Location using Content Relationship

列目では 2 ホップ先のホスト C, D のもつそれぞれのコンテンツのハッシュ値 ($\{0, 3, 7\}$ と $\{2, 5, 7\}$) のビットが立つ。従って、受信したクエリーを、最短ホップ数で目的のコンテンツまで到着するようなホストに決定的に転送することができ、局所的ネットワークでは高速な透過的コンテンツが実現できる。

4. コンテンツの関連性を利用した P2P 分散コンテンツ発見

4.1 コンテンツの関連性

ユビキタスコンテンツ環境では、共有されるコンテンツの数が膨大であるため、関連性を有するコンテンツも多数存在する。ここでは、P2P 広域分散ストレージにおいて複数のコンテンツへのアクセスの高速化を目的としているため、コンテンツの関連性を「同時にアクセスされる確率」と定義する。例えば、WWW における同一サイトのページ群や、大規模プログラムを構成するプログラムファイル群、ディレクトリ単位でダウンロードされるファイル群などが挙げられる。

P2P 広域分散ストレージシステム上で、このような関連性のある複数のコンテンツに対する同時アクセスを高速に行うためには、これらの複数のコンテンツを効率良く発見する必要がある。しかし、DHT ではコンテンツの関連性を考慮せず、個々のコンテンツを独立に発見する。そのため、発見する関連コンテンツの数が増加すれば全クエリーのホップ数の総和は線形に増加し、ネットワーク負荷の増加及びアクセス遅延の増大を招く。一方、Attenuated Bloom Filters は広域コンテンツ発見には適していない。

筆者らは、関連コンテンツ群への高速アクセスを実現するためには、広域コンテンツ発見機能と局所的コンテンツ発見機能の組合せが有効であると考えている。つまり、関連コンテンツ群をクラスタリングし、分散ストレージ上で互いに近傍に配置する。クラスタの発見には 3.2.1 で示した DHT のような最大ホップ数が保証された広域コンテンツ発見機能を利用する一方、クラスタ内のコンテンツ群の発見には 3.2.2 で示した Attenuated Bloom Filters のような高速なローカル発見機能を利用する。このようなコンテンツ発見手法を本稿では C-DHT (Cluster-based DHT) と呼ぶことにする。

4.2 C-DHT の概要

本節では、コンテンツの関連性を利用した高速な P2P 分散コンテンツ発見機構 C-DHT の基本コンセプトを示す。図 6 に示すように、C-DHT は以下の 4 つのプロセスから構成される。

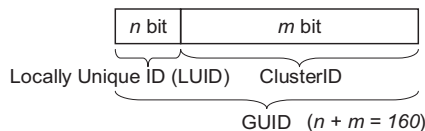


図 7 階層型 GUID

Fig. 7 Hierarchical GUID

(1) クラスタリング：共有コンテンツへのアクセスパターン[22]、ディレクトリ構造などに基づいてコンテンツをクラスタリングする。

(2) 近隣配置：4.3 で示す階層型 GUID を導入することにより、同一クラスタ内のコンテンツをオーバーレイ上で互いに近隣に配置する。これにより、クラスタ内でのコンテンツ発見にローカルコンテンツ発見手法を適用することが可能となる。

(3) 広域クラスタ発見：まず、クラスタ内の 1 つのコンテンツを広域発見機構により発見する。論理的に最大ホップ数が保証された DHT が利用できる。

(4) 局所的コンテンツ発見：広域クラスタ発見機構により発見したコンテンツを保持するノードを起点として、クラスタ内コンテンツをローカル発見機構により発見する。数ホップ以内に存在するコンテンツを高速に発見することが可能である Attenuated Bloom Filter [20] などが利用できる。

4.3 階層型 GUID

階層型 GUID は、図 7 に示されるように、クラスタ内でユニークな m bit の LUID (Locally Unique ID) とグローバルでユニークな n bit の ClusterID を用いて、合計 $(m+n)$ bit 表される ($\text{GUID} = \{ \text{LUID}, \text{ClusterID} \}$)。C-DHT におけるクラスタ発見には DHT として Tapestry [7] を用いることを想定している。Tapestry では NodeID のサフィックスの一致度に応じてオーバーレイネットワークが構築されるため、共通の ClusterID をもつコンテンツが近隣配置されるように、ClusterID をサフィックスとしてこれに LUID を結合して GUID を求める。本研究では、Chord [8] と同様に $m+n=160$ とする。

ClusterID, LUID はそれぞれ広域発見、局所的発見に利用される。ClusterID は、例えばディレクトリ構造によるクラスタリングを利用した場合にはパスのハッシュ値などを、アクセスパターンによるクラスタリングを利用した場合にはランダムな値などを用いる。一方、LUID は、例えばファイル名やコンテンツそのもののハッシュ値などを用いる。

4.4 課題

今後の検討事項としては以下のような点が挙げられる。

- クラスタ形成：分散ストレージ内に存在する全コンテンツを対象にクラスタリングするための分散的クラスタ形成機能が必要となる。

- クラスタ管理：クラスタにより含まれるコンテンツ数が異なることが考えられるため、クラスタのサイズ (m) が均一になるように、クラスタの合併及び分離機能が必要となる。

- 複製/キャッシュ管理：複製やキャッシュを DHT のようにコンテンツ単位で生成するのではなく、クラスタ単位で生成しなければならない。そのための複製/キャッシュ管理機構が必要である。

5. おわりに

本稿では、ユビキタスコンテンツ環境のための UCN の概要を解説するとともに、UCN の重要な機能である分散コンテンツ発見に関して、広域分散ストレージにおいて関連性のある複数のコンテンツに対する同時アクセスを高速化できるコンテンツ発見機

構の基本コンセプトを示した。

文献

- [1] M. Weiser. The computer for the twenty-first century. Scientific American, Sept. 1991.
- [2] Gnutella. <http://gnutella.wego.com/>.
- [3] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. Proc. ICSI Workshop on Design Issues in Anonymity and Unobservability, June 2000.
- [4] J. Kubiawicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. OceanStore: An Architecture for Global-Scale Persistent Storage. Proc. ASPLOS 2000, Nov. 2000.
- [5] Akamai. <http://www.akamai.com/>.
- [6] C. Plaxton, R. Rajaram, and A. Richa. Accessing nearby copies of replicated objects in a distributed environment. Proc. SPAA'97, June 1997.
- [7] B. Zhao, J. Kubiawicz, and A. Joseph. Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing. Technical Report, UCB/CSD-01-1141, April 2000.
- [8] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. Proc. ACM SIGCOMM 2001, Aug. 2001.
- [9] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-Addressable Network. Proc. ACM SIGCOMM 2001, Aug. 2001.
- [10] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. Proc. Middleware 2001, Nov. 2001.
- [11] S. Ratnasamy, S. Shenker, and I. Stoica. Routing Algorithms for DHTs: Some Open Questions. Proc. IPTPS '02, Mar. 2002.
- [12] W. Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley. The design and implementation of an intentional naming system. Proc. ACM SOSP'99, Dec. 1999.
- [13] Q. Lv, S. Ratnasamy, and S. Shenker. Can Heterogeneity Make Gnutella Scalable? Proc. IPTPS '02, Mar. 2002.
- [14] M. Balazinska, H. Balakrishnan, and D. Karger. INS/Twine: A Scalable Peer-to-Peer Architecture for Intentional Resource Discovery. Proc. Pervasive 2002, Aug. 2002.
- [15] C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen. Deployment Issues for the IP Multicast Service and Architecture. IEEE Network, Vol. 14, No. 1, Jan. 2000.
- [16] Y. Chu, S. G. Rao, and H. Zhang. A Case For End System Multicast. Proc. ACM SIGMETRICS 2000, June 2000.
- [17] 三村 和, 中内 清秀, 森川 博之, 青山 友紀. RelayCast: ピアツーピア型ストリーム配信のためのミドルウェア. 信学技報, IN2002-42, July 2002.
- [18] F. Dabek, M. Kaashoek, D. Kaeger, R. Morris, and I. Stoica. Wide-area cooperative storage with CFS. Proc. ACM SOSP'01, Oct. 2001.
- [19] A. Rowstron and P. Druschel. Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility. Proc. ACM SOSP 2001, Oct. 2001.
- [20] S. Rhea and J. Kubiawicz. Probabilistic Location and Routing. Proc. IEEE INFOCOM '02. June 2002.
- [21] D. Karger, E. Lehman, F. Leighton, R. Panigrahy, M. Levine and D. Lewin. Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web. Proc. ACM Symposium on Theory of Computing '97, May, 1997.
- [22] G. Kuenning and G. Popek. Automated Hoarding for Mobile Computers. Proc. ACM SOSP '97, Oct. 1997.