アプリケーション層マルチキャストにおけるアーカイブ取得方式の提案とその応用

坂野 遼平†　　佐藤 晴彦‡　　小山 聡‡　　栗原 正仁‡

北海道大学 工学部†　　北海道大学大学院 情報科学研究科‡

1. はじめに

近年、スケーラビリティや耐障害性に優れる Peer to Peer 型システムの研究が盛んである。本稿では、Peer to Peer 型のオーバーレイネットワーク上で実現される技術の 1 つであるアプリケーション層マルチキャスト（Application Layer Multicast, 以下 ALM とする）に焦点を当てる。

ALM はアプリケーションレベルでグループ管理を行い、1 対多通信を実現する技術であり、手法の提案[1]やミドルウェアの開発[2]等多くの研究が為されている。それらの研究成果は様々な大規模サービスに応用され得るが、構築するシステムによってはアーカイブ取得の需要が生じると考えられる。なお、本稿におけるアーカイブとはマルチキャストグループ内で過去にやり取りされたデータを指す。

しかしながら、ALM に関する従来の研究ではアーカイブの取得については触れておらず、アプリケーション開発者が実装上の工夫により実現する必要があった。そこで本研究では、ALM のアルゴリズムである Scribe [1]についてアーカイブ取得の効率的な手法を提案し、かつアプリケーション開発に有用な API を提供することを目的とする。また、実際の応用事例としてウェブ閲覧に対する横断的なコミュニケーションシステムの提案を行う。

2. 要素技術

2.1 Scribe

本研究の対象となる Scribe は、分散ハッシュテーブル（Distributed Hash Table, 以下 DHT とする）のアルゴリズムである Pastry [3]をベースとした ALM 手法である。DHT において、ある ID に対して複数のノードからルーティングを行うと、それらのルーティング経路の集合は木構造となる。Scribe では、これを配送木として利用している。配送木の根にあたるノードはランデブーポイント（Rendezvous Point, 以下 RP とする）と呼ばれ、RP に渡されたデータが順に子ノードに転送されていくことでマルチキャストが実現される。

2.2 Overlay Weaver

Overlay Weaver [2]は、Java 言語で実装された構造化オーバーレイのミドルウェアである。key-based routing [4]の概念に基づく階層化が為されているため、アプリケーション開発者にとっては利用するアルゴリズムを簡単に差し替えられる利点がある。またアルゴリズム研究者にとっては、ミドルウェアへのアルゴリズムの追加が容易であり、他の複数のアルゴリズムとの公正な比較を行える点が特徴である。

Overlay Weaver を構成する階層は、下位側からルーティング層、ハイレベルサービス層、アプリケーション層となっている。ハイレベルサービス層では DHT と ALM の 2 種類のインタフェースが提供されており、ALM の実装には 2.1 節で述べた Scribe が採用されている。

本研究の実装には、この Overlay Weaver を用いている。API の実装については 3.2 節で、応用システムの実装については 3.3 節で詳述する。

3. 提案手法

3.1 アーカイブ取得手法の提案

3.1.1 概要

本節では、アーカイブの取得について 3 種類の異なるアプローチを提案する。図 1 は各方式の概要を示したものである。

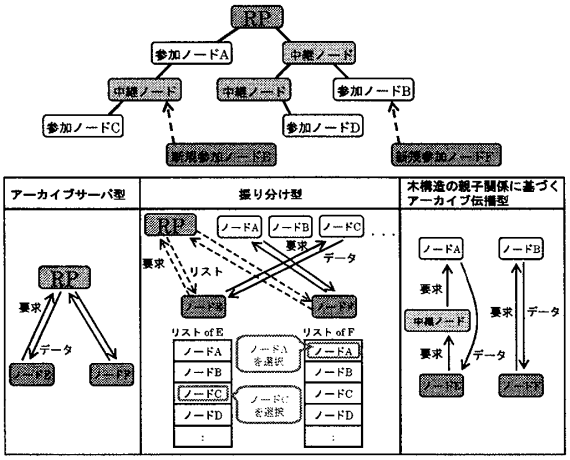

図 1 アーカイブ取得方式

3.1.2 アーカイブサーバ型

本方式では、グループの RP をアーカイブサーバに見立てて利用する。即ち、マルチキャストで送信されたデータを RP に集積しておき、必要に応じて RP からアーカイブの取得を行う。

アーカイブデータの取得/集積には DHT を用いる。本来、ALM と DHT は別個のサービス概念であるが、Scribe は構造化オーバーレイのノード関係をそのまま利用して配送木を構築するため、ALM に DHT の機能を並存させることが可能である。

DHT を用いることで実装時の見通しが良くなり、DHT や ALM に対する既存の研究成果を適用しやすくなる利点が生じる。例えば、アーカイブデータを扱う上での Churn 耐性については文献[5]の成果が適用できる。

3.1.3 振り分け型

ALM で扱うデータが大きい場合、前項の方式では RP

Acquisition of Archive Data in Application Layer Multicast and Its Applications.

† Ryohei BANNO; Faculty of Engineering, Hokkaido University

‡ Haruhiko SATO, Satoshi OYAMA, Masahito KURIHARA; Graduate School of Information Science and Technology, Hokkaido University

の負荷が大きくなってしまう。そこで本方式では、RPにはアーカイブを有するノードのリストを保持させておき、アーカイブサーバの役割を複数ノードに分散させる。

ノードリストの取得/更新には、前項と同様にDHTを用いる。アーカイブ取得要求ノードはRPからノードリストを取得し、リスト内からランダムにノードを選択してアーカイブを要求する。アーカイブを取得できた場合、自身をRPのリストに追加する。

3.1.4 木構造の親子関係に基づくアーカイブ伝播型

本方式では、アーカイブ取得に際しRPを経由せず、配送木の親子関係を利用する。グループに新たなノードが参加する場合、配送木上の親ノードを辿って最初に到達したグループ参加ノードからアーカイブを取得する。またグループに参加している各ノードは、マルチキャスト受信時にアーカイブの更新を行う。

グループへの参加ノードが多い場合、3.1.2項や3.1.3項の方式とは異なりRPに問い合わせが集中しないため、負荷が分散し可用性が向上すると考えられる。

3.2 提案手法の実装

3.1.2項で述べた提案手法に基づき、Overlay Weaverを拡張する形でAPIの実装を行った。具体的には、Overlay Weaverのハイレベルサービス層に、DHTとALMを並存させたMcastWithDHTというサービスを新たに追加し、アーカイブ取得可能なALMのインタフェースを構築した。

本APIを用いることで、アプリケーション開発者は通信処理の細部や経路表の保守といった下位層の処理を意識することなく、極めて容易にアーカイブ取得可能なALMシステムの開発を行うことができる。また、Overlay Weaver上に実装したことで、エミュレーションやネットワークの可視化、アルゴリズムの差し替えが可能であるという利点を受け継いでいる。

さらに本実装では、アプリケーション層においてMcastWithDHTShellというコマンドベースのサンプルプログラムも提供している。これをエミュレータと組み合わせることで、ルーティングアルゴリズムの動作試験や比較を容易に行うことができる。

3.3 提案手法の応用

実際の応用事例として、ウェブ閲覧に対する横断的なコミュニケーションシステムの開発を行った。本システムは、同一のウェブサイトを訪問しているユーザ同士でコミュニティを形成し、コメントを共有する機能を提供する。言わば、従来ウェブページ毎に設置されていたBBSやチャットの機能をブラウザ側に抽出するシステムである。

ウェブページ毎のコミュニケーションシステムとしては既存のサービス[6]も存在するが、それらはクライアント/サーバ型システムとして提供されており、サーバが単一障害点となる短所を持つ。Peer to Peer型のコミュニケーションシステムとしてはSkype [7]や各種インスタントメッセンジャーが挙げられるが、これらは予め知人であることが前提のコミュニケーションとなる。

本システムは、Peer to Peer型であるため単一障害点を持たず、また直接の知人で無くとも同じ場を訪れているという関係性のみによってグループを形成できる。

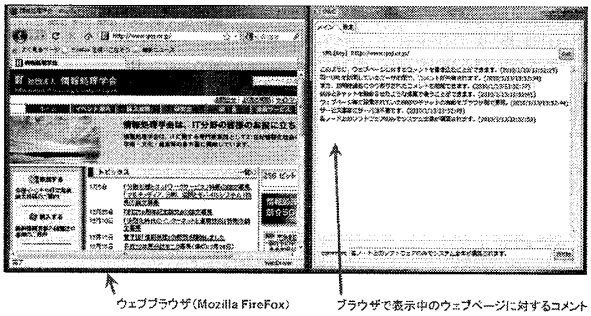

図2 提案システムの動作図

図2は本システムの動作の様子を示したものである。開発には、3.2節で実装したAPIを用いた。本システムでは、ウェブブラウザにおけるページ遷移に連動して自動的にURLを取得し、URL毎にグループを生成する。ウェブページを訪問したユーザは、アーカイブ取得機能により過去に交わされたやりとりを閲覧することができる。また、ウェブページに対するコメントを書き込むと、その時点で同じウェブページを閲覧している全ユーザにコメントがマルチキャストされる。

本システムにより、従来独立した行為であったウェブの閲覧に人々のリアルタイムなざわめきを導入することができ、特定のウェブページとの運営上の結びつきを持たない俯瞰的なコミュニケーションの場を提供することができる。

4. まとめ

本稿では、ALMアルゴリズムScribeにおけるアーカイブ取得手法を提案し、Java言語により提案手法のAPI及びサンプルプログラムを提供した。また、実装したAPIを用いてウェブ閲覧に対する横断的なコミュニケーションシステムの提案を行った。

今後の課題としては、3.1.3項及び3.1.4項で述べたアーカイブ取得手法の実装を行い、エミュレーションによって各手法の数値的な比較を行うことが挙げられる。多数のノードが高頻度にグループ遷移を行う状況等を模して、アーカイブの取得可能率やネットワークへの負荷等について評価を行いたいと考えている。

また3.3節で開発したシステムについては、NAT越えやセキュリティ等の面でより実用性を高められるよう改善を続ける予定である。

参考文献

[1] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron, SCRIBE: A large-scale and decentralised application-level multicast infrastructure, *IEEE JSAC*, Vol.20, No.8, pp.1489-1499, 2002
[2] 首藤一幸, 田中良夫, 関口智嗣, オーバレイ構築ツールキットOverlay Weaver, *Proc. SACSIS, pp.183-191,* 2006
[3] A. Rowstron and P. Druschel, Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems, *Middleware*, pp.329-350, 2001.
[4] F. Dabek, B. Zhao, P. Druschel, J. Kubiatowicz and I. Stoica, Towards a Common API for Structured Peer-to-Peer Overlays, *Proc. IPTPS*, pp.33-44, 2003
[5] 首藤一幸, 下位アルゴリズム中立なDHT実装への耐churn手法の実装, *Proc. ComSys*, pp.191-198, 2007
[6] Google SideWiki, http://www.google.com/sidewiki/
[7] Skype, http://www.skype.com/