

# P2P 型 Web 情報伝播モデルの提案と自律的リンク構築機構の実現

河木 孝治<sup>†</sup> 小瀬木 浩昭<sup>††</sup> 大矢 健太<sup>††</sup>  
鎌田 浩嗣<sup>††</sup> 村田 大樹<sup>†</sup> 武田 正之<sup>†</sup>

本稿では、公開された Web サーバ上の CGI プログラムを一つの Peer とみなし、各 Peer は互いにリンク情報を伝達しあい、キーワード情報を共有することで、情報同士の関係の自律的な構築と潜在的な共通事項の発見が可能な P2P 情報伝播モデルを提案する。また、共有情報に求められる速報性と網羅性を考慮しつつ、各 Peer への負荷の抑制を図る。さらに提案手法の応用例として、提案モデルを Wiki に適用した。これにより、通常一つの Wiki 内で閉じた自動リンクの関係を拡張し、複数の Wiki 間でキーワードを共有することが可能なシステムを実現した。

## Web base P2P Information Spread Model

KOUJI KAWAKI,<sup>†</sup> HIROAKI OZEKI,<sup>††</sup> KENTA OHYA,<sup>††</sup>  
HIROSHI KAMATA,<sup>††</sup> HIROKI MURATA<sup>†</sup> and MASAYUKI TAKEDA<sup>†</sup>

In this paper, each Peer considers CGI on the open to the public Web server to be one Peer, and proposes the P2P information spread model that autonomous construction of the relation of information and the discovery of a potential, common matter are possible by transmitting the link information each other, and sharing key word information. The proposal model was applied to Wiki, the relation of an automatic link that usually shut in one Wiki was enhanced, and the system that was able to share the key word between two or more Wiki was mounted.

### 1. はじめに

従来の Gnutella<sup>1)</sup>, JXTA<sup>2)</sup>, LimeWire<sup>3)</sup> などの実用的な P2P 技術は、Peer として主にクライアント PC を想定していた。そのため、Peer が頻繁に P2P ネットワークから切断される問題、プライベート IP アドレスや NAT により P2P ネットワークに参加できない Peer の問題、グローバルな IP アドレスを持っていたとしても動的な IP アドレスの変化への対応の問題など、様々な障害への対策が必要であり、クライアント PC で構成される P2P は効率があまり高くないという問題があった。また、共有資源として適切なものが見つからず、クライアント PC の HDD データなどが共有され、著作権法上の問題が発生した。また、一般の利用者が普段の用途にも利用するクライアント PC が Peer となる従来の P2P モデルは、セキュリティ上の危険性が高いという問題がある。あるクライアント PC が P2P ネットワークの一員になるには

P2P ネットワークの他の不特定多数の Peer に向け、グローバル IP アドレスを公開する必要があるが、近年、セキュリティ対策が十分に施されていないクライアント PC が、ウイルスやワームに感染し、多数の情報漏えい事件が発生し問題となっている。そのような事件を受け、クライアント PC にインストールする P2P ソフトの利用を禁止する企業や自治体も増加してきている。

Hybrid P2P に比べ、特別なサーバを必要としないという点で優位性のある Pure P2P であるが、例えば Gnutella のプロトコルは、他の Peer の情報を利用する際、情報が必要になってから検索クエリを送信するプル型モデルのため、他の Peer にある情報を探し当てるのに時間がかかるという点で、即時性において Hybrid P2P や通常の C/S 型モデルに比べ大きく劣っていた。

本稿では、公開された Web サーバ上の CGI を一つの Peer とみなし、各 Peer は互いにプッシュ方式でリンク情報を伝播しあい、情報同士の関係を自律的に構築することで、P2P ネットワーク内での情報共有が可能な、Pure P2P 型の情報伝播モデルを提案する。提案手法は、情報発見の即時性と網羅性の確保を、構成 Peer への負荷を一定の範囲におさえた形で提供

<sup>†</sup> 東京理科大学理工学部情報科学科  
Dept. of Information Sciences,  
Tokyo University of Science

<sup>††</sup> 東京理科大学大学院理工学研究科情報科学専攻  
Graduate School of Science and Technology,  
Tokyo University of Science

可能にする。また、一定期間おきに処理を行う定期処理プロトコルの導入により、Pure P2P の構成を保ったまま、P2P ネットワーク内での情報共有が可能になるという特徴を持つ。提案モデルを Wiki<sup>4)</sup> に適用した。共有情報は「キーワードと対になるハイパーリンク」であり、リンク情報の伝播で情報共有を実現する。Wiki のページ名をキーワードとし、Wiki ページの URI をキーワードと結び付けられるリンクとする。これにより、通常一つの Wiki 内で閉じた名前空間を拡張し、複数の異なる Wiki 間で、Wiki のページ名である WikiName を共有することが可能なシステムを実現した。さらに、拡張された Wiki とブログを結びつけることで、ブログ中に登場したキーワードへの自動的なリンク付け機構の実装について紹介する。

提案モデルは、既存の公開 Web サーバ上の CGI を Peer とする点で、既存のクライアント PC を Peer とするモデルに比べ、ファイアウォールなどの追加のセキュリティ対策が少なく済むため安全性に優れている。共有手段として、既存の公開 Web コンテンツを発見するリンク情報を伝播させるため、P2P 型ファイル共有ソフトなどで行われる、共有のためのコンテンツの複製が発生せず、著作権上の問題の発生を防止できる。Peer である公開された Web サーバ上の CGI は、静的で一意なグローバル IP アドレスと URI を持ち、安定した回線品質を持つという点で、既存のクライアント PC を Peer とする P2P モデルに比べ、障害対策の簡易化ができ、効率的である。また、情報到達の信頼性の点でも優位である。

提案モデルは、例えば、複数のブログ間での自動的な言及リンクの生成や、分散型の電子的な百科辞典を構築する試みへの応用など、汎用的で様々な応用の可能性が考えられる。近年、Web2.0<sup>5)</sup> という、次世代型のインターネットやインターネットサービス、あるいはそれを構成する仕様の事をさす言葉が登場している。要約すると、従来の、静的な HTML で構成された Web (Web1.0)、CGI や CMS を積極的に取り入れ HTML の大半が動的に生成される Web (Web1.5) に対して、Web1.0 の静的な Web であつたり、Web1.5 のサイトの独立性が高いダイナミックな Web とは異なり、サーバやコンテンツ同士がシームレスに連動され、インターネットが社会的なネットワークとして動作するという様な意味合いを示す言葉である。インターネット全体がコミュニケーションプラットフォームとして連動、共鳴し、進化する Web。具体的には、サイトやサービス、コンテンツがコンポーネント化され、コンポーネント化されたリソースがインターネットと言うインフラの上で複合的に利用される世界である。本稿の提案は、Web2.0 における基盤の一部となりえる可能性を秘めている。

## 2. 本提案モデル

提案モデルは、サーバなど、他の Peer を管理する役割を持つような、特別な役割を持つ Peer を必要とせず、構成 Peer 同士がすべて対等な関係を持つ、Pure P2P 型のネットワーク構成をとる。

### 2.1 提案モデルと構成要素

提案モデルの構成要素を図 1 に、キーワード作成から参照リンク形成までの流れを図 2 に示す。

提案モデルは、次の 3 層で構成される。

**Web 層：**

Web サーバ上の CGI として動作している Wiki、Blog、CMS (Contents Management System) などからのキーワード作成と利用を行う層。

**P2P フレームワーク層：**

キーワード情報など情報の伝播を行う層。

**ベース層：**

一意な URI を保証しフレームワークの動作とデータベースを保管する層。

複数の Peer 間でのキーワード伝播などの通信を行うのは、P2P フレームワーク層が担当する。

新規キーワード作成から通知までの流れ：

(1) CGI 上で動作している Wiki や Blog などから、利用者の明示的な操作、あるいは自動生成によりキーワードが生成・抽出される、(2) 生成されたキーワードは P2P フレームワーク層によって他の Peer へ伝播を行う、(3) 伝播されたキーワードはベース層のデータベースによって、キーワードリストとして保管される。

文章作成から参照リンク形成までの流れ：

(1) 利用者が Wiki や Blog のあるページを執筆する

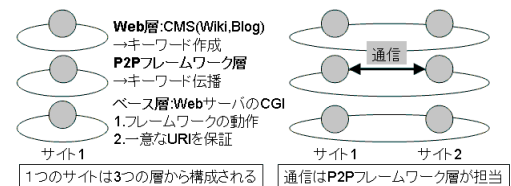


図 1 提案モデルの構成要素  
Fig. 1 Component of proposal model

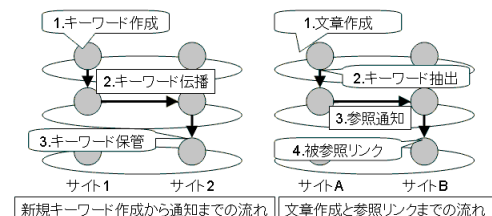


図 2 キーワード作成から参照リンク形成までの流れ  
Fig. 2 Flow from key word making to reference link formation

と、(2) その文章と、ベース層に蓄積されているキーワードリストを比較し、一致するキーワードを抽出する。(3) 抽出されたキーワードは P2P フレームワーク層へ渡され、そのキーワードへの言及が行われた旨を示すキーワード参照通知を、そのキーワードを最初に作成した Peer へ通知する。(4) 通知された Peer は、そのキーワードへ言及した Peer として、利用者がそのキーワードを言及したページの URI へのリンクを生成し、自身のキーワードのページへ、言及ページへのアドレスという表現で掲載する。

## 2.2 本提案モデルの特徴

### 2.2.1 Peer は公開された Web サーバ上の CGI

従来の実用的な P2P 技術は、Peer として主にクライアント PC を想定していた。そのため、動的な IP アドレスの変化、クライアント PC のシャットダウンなどにより Peer が P2P ネットワークから不意に消えてしまう頻度が高くその点がシステムのボトルネックとなっていた。そのためシステムには冗長化が不可欠であり、過剰な冗長化により無駄なコストが生じる問題があった。また、プライベート IP アドレスや NAT の問題により、P2P ネットワークと通信するために、中継 Peer が必要であった。

提案モデルでは、Peer として公開された Web サーバ上の CGI を採用することで、Peer は常に静的で一意、グローバルな URI と安定したネットワーク回線を確認することが可能となる。P2P ネットワーク内の Peer に一定の安定性が確保されることで、これまでボトルネックとなっていた Peer の定期的また頻繁な生存確認や冗長化のコストを低く抑えることが可能となる。

### 2.2.2 P2P ネットワーク内の全 Peer の URI を保持

従来の Gnutella などの P2P ネットワークでは、P2P ネットワークのトポロジは固定的で、一度決定されたら再接続されるまで変化することは無かった。そのため、仮に、条件に当てはまるいくつかの Peer にだけメッセージを送りたい場合も、P2P ネットワーク全体に対してブロードキャストする必要があった。

提案モデルでは、新規に P2P ネットワークに参加する Peer は自身の URI を全 Peer に向けて 1 回だけ送信することで、P2P ネットワーク内の Peer が常に他の全ての Peer の URI を保持していることを高い確率で保証する。これにより、伝達プロトコルなどを用いて効率的に特定メンバへの同報配信を行うことが可能となる。また、メッセージの種類や要求品質に応じて、メッセージ伝達の際のネットワークトポロジを任意に設定することが可能となり、従来に比べより柔軟な品質要求でのメッセージ処理が可能となる。

### 2.2.3 Peer 間でキーワードという情報を共有

従来の P2P ネットワークでは、情報が必要になってから検索するため、仮に P2P ネットワーク内に有用な情報が存在しても、発見までに時間がかかるという問題があった。また、例えば Gnutella では、多数の検索クエリによる P2P ネットワークにかかる負荷を防止するため、検索クエリの届くホップ数を制限していた。そのため、検索クエリの届かない Peer 中の情報を検索することができない制限が生じていた。

提案モデルでは、各 Peer はそれぞれ複数の「キーワード」を持っている。キーワードが追加・削除されると、伝達プロトコルにより、各 Peer にプッシュ方式で配信される。この仕組みにより、P2P ネットワークに速報性を持たせることが可能となる。また、それぞれの Peer は P2P ネットワーク内のその他の Peer の URI を保持することを高い確率で保証することで、Pure P2P 環境下において、検索の網羅性を確保することが可能となる。

## 2.3 伝達プロトコル

伝達プロトコルの速報性と網羅性についての概念を図 3 に、網羅・遅延型伝達プロトコル処理の例を図 4 に、部分・速報型伝達プロトコル処理の例を図 5 に示す。大きく 3 つの伝播の範囲概念がある。

### (a) 網羅性が求められる情報伝播

P2P ネットワークへの Peer の参加・離脱通知などは、速報性よりも、他の全ての Peer へ通知するという高い網羅性が求められる。そのため、全 Peer へ通知されるよう伝播を依頼していき、依頼された各 Peer では定期的（1日に数回から数日に 1 回程度の頻度）に処理がされる。これにより、広くゆったりとほぼ全ての Peer へ情報が伝播される。

### (b) 速報性が求められる情報伝播

時事的なキーワードの追加など要求される速報性の高い通知は、即時に処理・伝播させる代わりに、少ないホップ数かつ、一定期間に 1 つの Peer が作成できる数を制限し、P2P ネットワーク全体にかかる負荷を抑える。

### (c) 伝播情報に応じた柔軟な情報伝播

通常のキーワードの追加通知などは、伝播される内容により、処理頻度や経由可能なホップ数を変化させる。即時性が低いキーワードは、伝播処理の頻度を低く抑える代わりに、経由可能なホップ数を多く設定し広範に伝播することを認める。即時性が高いキーワードは、経由可能なホップ数を少なく抑える代わりに、伝播処理の頻度を高く設定する。

## 3. 提案モデルの Wiki への適用

### 3.1 概要

Wiki とは Web ブラウザから簡単に Web ページ

の発行・編集などが行なえる，Web コンテンツ管理システムである．1つのWiki サイトは複数のWiki ページで構成され，1つのWiki ページは通常一意なURIを持ち，WikiName形式（2文字以上の大文字が含まれる英単語）のページ名とページ本文で構成される．Wiki上のページ名であるWikiNameの名前空間

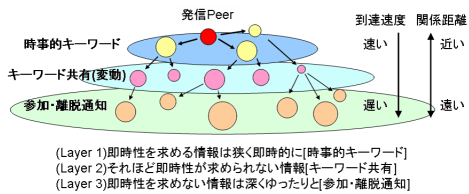


図3 伝達プロトコルの速報性と網羅性  
 Fig. 3 News flash and covering of transmission protocol

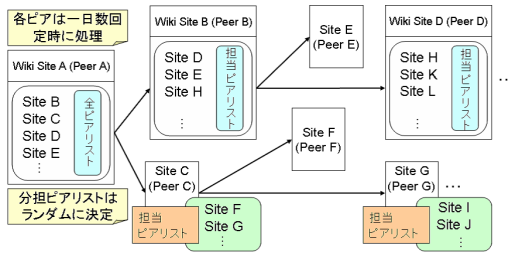


図4 網羅・遅延型伝達プロトコルの処理例  
 Fig. 4 Example of processing covering and delayed transmission protocol

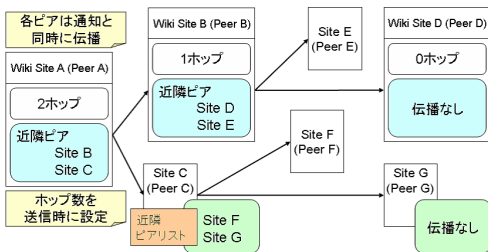


図5 部分・速報型伝達プロトコルの処理例  
 Fig. 5 Example of part and news flash type transmission protocol processing

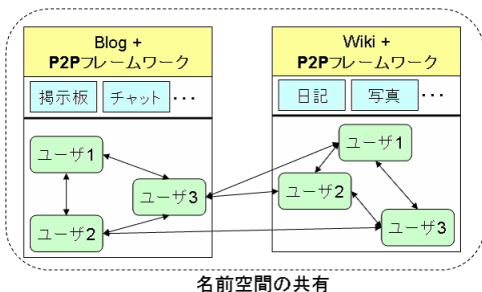


図6 名前空間の共有  
 Fig. 6 Sharing of name space

は通常1つのWiki サイト内に閉じており，異なる場所に設置したWikiName同士が自動的に結びつくことはない．

そこで，P2P フレームワークにおいての共有キーワードとしてWikiNameを設定することで，複数のWiki間でWikiNameの名前空間が共有できる「WikiName共有」を実現する(図6)．

Webサーバ上のCGI上で動作するWikiを一つのPeerとみなし，全PeerでWikiNameを共有するサービスを実現する．Peerが，キーワードとそれに対応するページを作成すると，キーワード情報がサービスに参加する全てのPeerで共有される．また，各Peerの文章中でWikiNameが出現した場合，そのWikiNameのページへ自動的にリンク付けされる(図7)．

### 3.2 システムの動作概要

システムの動作概要を図8に示す．

(1)Peer Aが，キーワード「」に対応するページを新規作成すると，(2)そのキーワード情報が他のPeerへ伝播する．(3)Peer Bが「」について自身のWikiサイト内のページ中で言及すると，「」に対応するページに自動的にリンク付けされ，また言及通知がPeer Aに送信され，Peer A上の「」に対応するページに，そのページの言及ページとしてPeer Bの該当ページへのリンクが掲載される．この仕組みによりWebページ同士の自律的な関係構築が可能となる．

### 3.3 実現システムの詳細

#### 3.3.1 表記方法

主体Xの公開鍵を $P(X)$ ， $P(X)$ に対応する主体Xの秘密鍵を $S(X)$ と表記する．証明書の内容を $\langle$

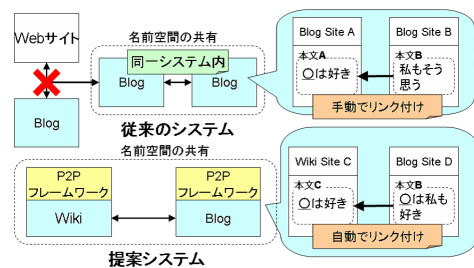


図7 自動的なリンク付け  
 Fig. 7 Automatic link putting

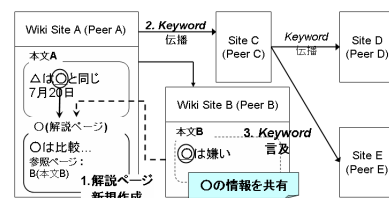


図8 システムの動作概要  
 Fig. 8 Outline of operation of system

で括弧で表現し、その証明書に  $S(X)$  で電子署名が施されていることを、 $\langle \dots \rangle S(X)$  と表現する。主体  $Y$  のネットワーク上での識別子を  $URI(Y)$  と表記する。Wiki ページ  $page$  のネットワーク上での識別子を  $URI(page)$  と表記する。通知文を [通知内容] と表記する。また、紙面の都合上、 $\{a \in A, b \in B; (a, b)\}$  を略記して  $\{a, b\}$  と表記しているところがある。

### 3.3.2 Peer

各 Peer は Web サーバ上で動く一つの CGI プログラムであり、通常オンラインの環境を想定し、静的で一意的なグローバル IP アドレスと DNS による URI の対応を持つ。また、セキュリティ確保のため、Peer 毎に専用の、PKI に基づく公開鍵と秘密鍵の組を生成、保持する。

### 3.3.3 各 Peer の保持する情報

全ての Peer の集合を  $U$ 、 $U$  の部分集合を  $G \subseteq U$  とする。 $G$  は  $U$  のほとんど全ての要素を含む部分集合とする。共有キーワードの集合を  $K$  とする。各 Peer はそれぞれ以下の情報を保持する。

(1) Peer リスト:  $\{\forall x \in G; URI(x), P(x)\}$

グループ  $G$  に含まれる Peer の URI、その Peer の公開鍵、の組みのリスト。  $PList$  と略記する。自身の Peer 情報はこのリストへ含めない。

(2) Keyword リスト:

$\{\forall k \in K, \exists x \in G; k, URI(page), URI(x)\}$

共有情報であるキーワード、そのキーワードに対応するページの URI、その対応するページを持つ Peer の URI、の組みのリスト。  $KList$  と略記する。

(3) Keyword 参照リスト:  $\{\exists k \in K; k, URI(page)\}$

自サイトから参照を行っているキーワード (自身が作成したキーワードも含む)、そのキーワードへ参照を行っている自サイトのページの URI、の組みのリスト。  $KRefList$  と略記する。

(4) Keyword 被参照リスト:

$\{\exists k \in K; k, URI(page)\}$

参照が行われている自サイトのキーワード、そのキーワードへ参照を行っているページの URI (自サイト内のページの URI も含む)、の組みのリスト。  $KRefdList$  と略記する。

## 3.4 基本プロトコル

各通知は、ヘッダ部とメッセージ部から構成され、ヘッダ部は  $(URI(差出 Peer), URI(宛先 Peer))$  から構成される。

### 3.4.1 Peer の参加

(1) 新規に P2P ネットワークに参加する Peer  $A$  は、既に参加しているいずれかの Peer (ここでは Peer  $B$  とする) から  $PList, KList, P(B)$  を受け取り、 $PList$  に  $(URI(B), P(B))$  を追加する。

(2) 参加通知として、

$\langle$ ヘッダ部; [参加通知],  $URI(A), P(A)\rangle$  を、 $\forall peer \in PList$ , に送信する。

(3) 参加通知を受け取った各 Peer は、 $PList$  に Peer  $A$  に関する情報  $(URI(A), P(A))$  を追加する。

### 3.4.2 Peer の脱退

(1) サービスから脱退する Peer  $C$  は、脱退通知として、 $\langle$ ヘッダ部; [脱退通知],  $URI(C)\rangle S(C)$  を、 $\forall peer \in PList$ , に送信する。

(2) 脱退通知を受け取った各 Peer は、受け取った電子署名を保管してある  $P(C)$  を用いて検証し、正当性を確認した後、 $PList$  から Peer  $C$  に関する情報  $(URI(C), P(C))$  を削除する。以下、各通知は同様に正当性の検証を行うものとする。

### 3.4.3 キーワードの追加

(1) キーワード  $NewKeyword$  の追加を行う Peer  $D$  は、自サイト内に、 $NewKeyword$  と本文から構成されるページを作成する。この  $NewKeyword$  はページ名 (WikiName) となる。

(2) 自身の  $KList$  に

$(NewKeyword, URI(NewKeyword), URI(D))$

を追加する。

(3) キーワード追加通知として、

$\langle$ ヘッダ部; [キーワード追加通知],  $NewKeyword, URI(NewKeyword)\rangle S(D)$  を

$\forall peer \in PList$  に送信する。

(4) キーワード追加通知を受け取った各 Peer は、 $(NewKeyword, URI(NewKeyword), URI(D))$  を  $KList$  に追加する。

### 3.4.4 キーワードの削除

キーワードの削除はそれに対応するページを持つ Peer のみが行えるとする。

(1) キーワード  $RmKeyword$  の削除を行う Peer  $E$  は、自身の  $KList$  から

$(RmKeyword, URI(RmKeyword), URI(E))$

を削除する。

(2)  $\exists page,$

$(RmKeyword, URI(page)) \in KRefList$ , ならば

そこから  $(RmKeyword, URI(page))$  を削除する。

(3) 同様に自身の  $KRefdList$  から

$(RmKeyword, URI(page))$  を削除する。

(4) キーワード削除通知として、

$\langle$ ヘッダ部; [キーワード削除通知],  $RmKeyword, URI(RmKeyword)\rangle S(E)$  を

$\forall peer \in PList$  に送信する。

(5) キーワード削除通知を受け取った各 Peer は、 $(RmKeyword, URI(RmKeyword), URI(E))$  を  $KList$  から削除する。

(6)  $\exists page,$

$(RmKeyword, URI(page)) \in KRefList$ , ならば

そこから  $(RmKeyword, URI(page))$  を削除する .

### 3.4.5 キーワード参照追加

Peer  $F$  が  $page$  を編集するとき ,

$\exists Keyword \in page \wedge KList$  , ならば次のことを行う .

(1)  $Keyword$  に対応するページ  $URI(Keyword)$  へのリンク付けを行う . この  $Keyword$  に対応するページは複数ある場合もある .

(2)  $(Keyword, URI(page))$  を  $KRefList$  に追加する .

(3) キーワード参照追加通知として ,  $\langle$ ヘッダ部; [キーワード参照追加通知] ,  $Keyword, URI(page)\rangle S(F)$  を ,  $\exists peer \in PList$  ,

$(Keyword, URI(Keyword), URI(peer))$

$\in KList$  , に送信する . 1つのキーワードに対して , 複数のキーワードに対応するページが存在した場合には , その全てへ通知を送信する .

(4) キーワード参照追加通知を受け取った各 Peer は  $KRefList$  に  $(Keyword, URI(page))$  を追加する .

### 3.4.6 キーワード参照削除

Peer  $G$  が  $page$  を削除するとき ,

$\exists Keyword \in page \wedge KList$  , ならば次のことを行う .

(1)  $(Keyword, URI(page))$  を  $KRefList$  から削除する .

(2) キーワード参照削除通知として ,  $\langle$ ヘッダ部; [キーワード参照削除通知] ,  $Keyword, URI(page)\rangle S(G)$  を  $\exists peer \in PList$  ,

$(Keyword, URI(Keyword), URI(page))$

$\in KList$  , に送信する . 1つのキーワードに対して , 複数のキーワードに対応するページが存在した場合には , その全てへ通知を送信する .

(3) キーワード参照削除通知を受け取った各 Peer は  $KRefList$  から  $(Keyword, URI(page))$  を削除する .

## 3.5 応用プロトコル

### 3.5.1 簡易障害検知プロトコル

ネットワーク障害で宛先サーバに到達できない場合や , 宛先サーバの障害などが原因で , そもそも P2P フレームワーク層までメッセージが到達できない場合がある . 早期な障害検出のために , メッセージ送信の前にサーバへの簡単な生存確認 ping を送ることで , 予め障害のあるサーバを避けてメッセージを伝達することを可能にする .

### 3.5.2 擬似 cron 処理

Webサーバによっては , CGI プログラムに対する cron ( 定期作業を実行するデーモン ) が利用できない場合がある . 回避措置として , 他からの任意の Web ページへのアクセスに対して , 連動して同時に CGI を実行し , cron 処理が使えない Web ホスティングサーバなどでも擬似的な定期処理を実現した .

## 4. 実装・動作例 ( 実行例 )

### 4.1 処理系

P2P 情報伝播のフレームワークは Perl5<sup>6)</sup> を利用して実装した . Webサーバには Apache1.3.29<sup>7)</sup> を利用した . P2P 情報伝播のフレームワークと連携する Wiki は PukiWiki1.4.5<sup>8)</sup> を , 連携するブログは tDiary 2.0.2<sup>9)</sup> をベースにした .

### 4.2 実装について

#### 4.2.1 実装システムの構成

実装システムの全体構成は図 1 に示した 3 層で構成され , その中でキーワード伝播を担う P2P フレームワーク層は , アプリケーション部 , 通信マネージャ部 , 通信部 , の 3 部で構成される ( 図 9 ) . 各部について解説する ( 図 10 ) . アプリケーション部は , Web 層からのデータを受け取り , キーワードの抽出や , P2P ネットワーク維持のためのコマンドを通信マネージャ部へ渡す . 通信マネージャ部は , Peer 間の通信の効率化を担う . 1対多の転送要求の 1対1の転送要求への変換などを行い , 通信部に渡す . 通信部は , 1対1の Peer 間の通信を行う . PKI を用いたメッセージの改ざんの検出などもこの層で行う .

#### 4.2.2 実装上の工夫

P2P フレームワーク層は , Web 層の CMS ( Wiki , Blog ) などの個別のシステムに依存しないよう実装を工夫した ( 図 11 ) . 本システムが CMS と連携するとは記事の編集に応じて自動的にキーワードの処理を行うということであり , そのためには CMS 上での記事の編集や表示等のイベントを取得し , 記事の内容

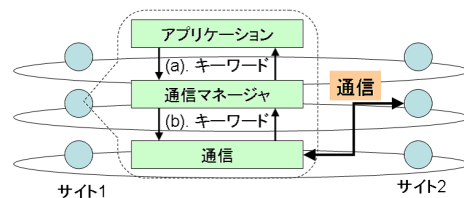


図 9 P2P フレームワーク層の概要  
Fig. 9 Outline of P2P framework layer

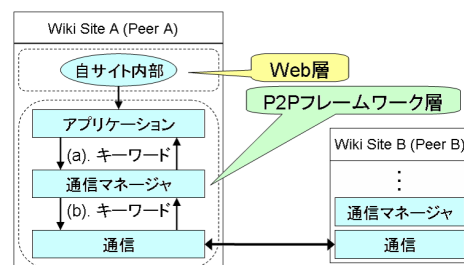


図 10 P2P フレームワーク層の詳細  
Fig. 10 Details in P2P framework layer



を解析する必要がある。今回の実装では PukiWiki と tDiary との連携を実現しているが、その他のシステムとの連携も容易に実装出来るように各 CMS に合わせて実装する部分を最小化して構成している。各 CMS 毎に実装する機能は、上記イベントを取得し、その記事を本システムに通知、応答で語句と URI の関係を取得し、それを記事へリンクとして反映することに絞られており、記事の内容を解析し、キーワードデータの状況とあわせて適切な処理を呼び出すことは、共通部分として本システムに含んで実装した。例えば対象の CMS にプラグイン機能があればプラグインとして実装することが出来、CMS の本体コードを変更できる場合はより一体化した形で実装することが可能である。今回の実装対象の PukiWiki, tDiary はどちらもプラグイン機能をもつが、PukiWiki ではプラグインでの実装と本体コードの変更を併用し、ユーザが通常の操作だけで利用出来るようになっており、tDiary ではプラグインでの実装のみを行い、ユーザはプラグイン呼び出しの操作により利用出来るようになっている。なお、CMS とどのように連携するかは対象の環境に応じて選択することが可能である。

#### 4.3 動作例

動作例として実行画面のスクリーンショットを図 12, 図 13, 図 14 に示す。図 12, 図 13 では、サイト外への自動リンク機能が「[0]」と表示されている。1つのキーワードに対応するページが複数存在すれば、[0], [1]... とリンク付けされる。また、JavaScript と CGI を組み合わせて、本フレームワークに対応していない、一般のブログでのキーワード共有のための自動リンクの仕組みを実現した(図 14)。これは、キーワードリストを本フレームワーク対応のサーバから取得し、取得したキーワードリストと、ブログ中に登場する単語の合致を判定し、リンクの自動推薦を実現する。

### 5. 提案システムにおける負荷の傾向予測

実現システムの実測データと類似する、はてなダイアリー<sup>11)</sup>と日本語版 Wikipedia<sup>12)</sup>を調査して得られたデータから、提案システムにおける負荷の傾向を予測する。

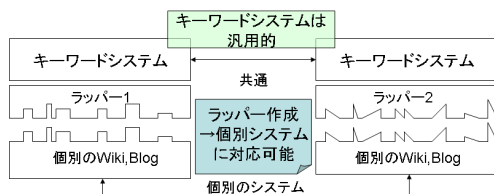


図 11 P2P フレームワークの汎用化  
Fig. 11 Generalizing of P2P framework

#### 5.1 通信における負荷

各 Peer 同士で通信が発生するのは、任意の Peer の参加・脱退、任意の Peer のキーワード作成・削除、自 Peer のキーワード参照追加・参照削除、自キーワードの参照追加・削除、の各処理の際である。各処理の発生頻度と処理 1 回あたりの通信回数の積の和が、P2P ネットワーク全体の通信頻度である。

#### 5.2 はてなダイアリーに対する調査

##### 5.2.1 類似のシステムの傾向データ

実現システムの通信頻度の計算におけるパラメータを決定するため、はてなダイアリーについて調査した。2005 年 9 月現在公開されているデータによると、日記数は 19 万件、キーワード数は 12 万 6 千個であり、最近 16ヶ月で共にほぼ線形に、日記は 13 万 8 千件、キーワードは 8 万 4 千個増加している。最近更新された日記を 50 件おきに調べたところ、1 記事の平均



図 12 Wiki 上で動作している様子  
Fig. 12 Appearance of operation on Wiki.

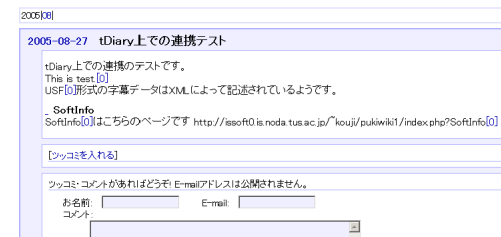


図 13 tDiary 上での連携の様子  
Fig. 13 Appearance of cooperation on tDiary

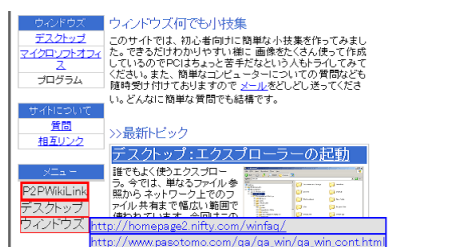


図 14 JavaScript を利用した一般サイトへの自動リンク付け  
Fig. 14 Automatic link putting on general site using JavaScript

キーワード言及数は 4.05 個，1 日記の 1 日の平均記事数は 1.31 個であった．なお，日記の減少数，記事やキーワードの削除件数は不明だが，増加数に比べれば無視できる程度の数と考えられるため，今回は考慮から外した．

### 5.2.2 負荷傾向の推定

Peer 数と通信回数(回/日)の関係は図 15 のようになった．1 日の平均新規参加 Peer 数は 283.95 個 (= 最近 16ヶ月間の日記増加数 / 最近 16ヶ月間の日数 = 138000/486)，1 日の平均キーワード増加数は 172.84 個 (= 最近 16ヶ月間のキーワード増加数 / 最近 16ヶ月間の日数 = 84000/486)，1 日の 1 Peer の平均キーワード参照数は 5.32 件 (= 1 記事の平均キーワード言及数 × 1 日記の 1 日の平均記事数 = 4.05 × 1.31)，1 日の 1 Peer の平均自キーワード被参照数は 5.32 件 (= 1 記事の平均キーワード言及数 × 1 日記の 1 日の平均記事数 = 4.05 × 1.31) である．よって，通信頻度は 1 日に平均して 640.27 回 (= 1 日の平均新規参加 Peer 数 + 1 日の平均自キーワード増加数 × 自分以外の Peer 数 + 1 日の平均他キーワード増加数 × 1 + 1 日の 1 Peer の平均キーワード参照数 + 1 日の 1 Peer の平均自キーワード被参照数 = 283.95 + 172.82 × 1/190000 × 189999 + 172.82 × 189999/190000 × 1 + 5.32 + 5.32) となる．つまり，1 Peer あたり 2 分に 1 回程度の通信頻度になると推定される．また日記数とキーワード数は共に線形に増加していることから比例関係にあり，キーワード数と 1 記事あたりのキーワード言及数も，キーワードが多いほど記事中に含まれるキーワードも多くなると考えられることから比例関係にあるものとして，Peer 数が 2 の場合と現在の倍の 380000 の場合も推定した．

多くのキーワードはコンスタントに言及されており，例えば，はてなダイアリーの 2005 年 9 月 24 日現在の最も言及されているキーワードである「orz」は 1 日約 750 回の言及数であった．時事的なキーワードとして一時的な言及数の多い「台風」は，過去 3 ヶ月間のうちの最大が 4500 件，「地震」は 3800 件の言及

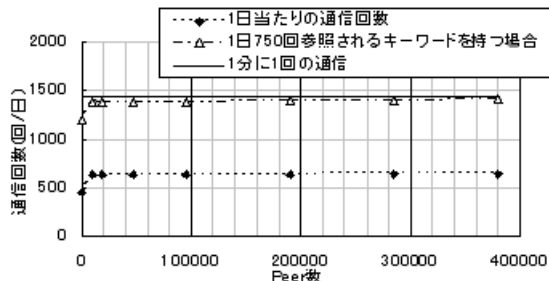


図 15 日記数 (Peer 数) と通信回数の関係

Fig. 15 Relation between number of diaries(number of peer) and communication frequency

数であった．仮に最大 1 日に 5000 回の言及数のあるキーワードを持った場合，通常の負荷と合わせると 1 日に 5640 回，1 分間平均で 4 回程度の通信頻度となるが，この程度の負荷であれば通常の利用の範囲内と考えられるため，実用上は特に支障はないと思われる．

## 5.3 Wikipedia に対する調査

### 5.3.1 類似のシステムの傾向データ

実現システムの通信頻度の計算におけるパラメータを決定するため，日本語版 Wikipedia について調査した．2005 年 6 月現在公開されているデータによると，執筆者・編集者であるウィキペディアンは 3023 人，記事数は 12 万 7 千件，内部リンク数は 270 万個であった．最近 16ヶ月で，共にほぼ線形に，ウィキペディアンは 2365 人，記事数は 9 万 4 千個増加している．よって，1 記事の平均内部リンク数は 21.26 個 (= 内部リンク数 / 記事数 = 2700000/127000)，ウィキペディアン 1 人が執筆する 1 日の平均記事数はウィキペディアンが 3023 人の時点では 0.0640 件 (= 最近 16ヶ月間の記事増加数 / 最近 16ヶ月間の日数 / ウィキペディアン数 = 94000 ÷ 486 ÷ 3023) となる．なお，ウィキペディアンの減少数，記事の削除件数は不明だが増加数に比べれば無視できる程度の数と考えられるため，今回は考慮から外した．

### 5.3.2 負荷傾向の推定

Peer 数と通信回数(回/日)の関係は図 16 のようになった．ただし，ウィキペディアンを Peer，記事の項目名をキーワード，内部リンクをキーワード参照と対応させている．1 日の平均新規参加 Peer 数は 4.87 個 (= 最近 16ヶ月間のウィキペディアンの増加数 / 最近 16ヶ月間の日数 = 2365/486)，1 日の平均キーワード増加数は 193.42 個 (= 最近 16ヶ月間の記事増加数 / 最近 16ヶ月間の日数 = 94000/486)，1 日の 1 Peer の平均キーワード参照数は 1.36 件 (= 1 記事の平均内部リンク数 × その時点でのウィキペディアン 1 人が執筆する 1 日の平均記事数 = 21.26 × 0.0640)，1 日の 1 Peer の平均自キーワード被参照数は 1.36 (= 1 記事の平均内部リンク数 × その時点でのウィキペディアン 1 人が執筆する 1 日の平均記事数 = 21.26 × 0.0640

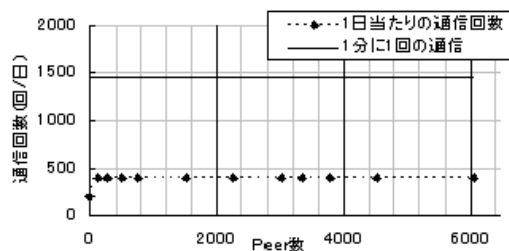


図 16 ウィキペディアン数と通信回数の関係

Fig. 16 Relation between number of wikipedian and communication frequency



)である。よって、通信頻度は1日に平均394.29回(=1日の平均新規参加Peer数+1日の平均自キーワード増加数×自分以外のPeer数+1日の平均他キーワード増加数×1+1日の1Peerの平均キーワード参照数+1日の1Peerの平均自キーワード被参照数=4.87+193.42×1/3023×3022+193.42×3022/3023×1+1.36+1.36)となる。つまり、1Peerあたり3分に1回程度の通信頻度になると推定される。また、ウィキペディアン数と記事数は共に線形に増加していることから比例関係にあり、キーワード数と1記事あたりのキーワード言及数も、キーワードが多いほど記事中に含まれるキーワードも多くなると考えられることから比例関係にあるものとして、Peer数が2の場合と現在の倍の6046の場合も推定した。

#### 5.4 負荷要因からの応用システムの条件の推測

調査の結果から、実現システムの負荷のうち大きな要因となるのは、1日のキーワード増加数である。このパラメータは2つの調査結果のいずれでもPeer数に比例せず、ほぼ定数となっている。このため実現システム全体の負荷もPeer数の増加に対して増えにくくなっており、現実的なPeer数において実用上問題ない値となっている。つまり1日のキーワードの増加数が一定となることが実現システムを応用する際に実用性を保つための条件と考えられる。

#### 5.5 実用性に問題のある応用システム例

実用性に問題のある応用例として、各日記サイトを1つのPeerとし、日記中の個々の記事のタイトルを共有するシステムが考えられる。記事は各Peerである程度の頻度で書かれ続けるため、1日の記事増加数はPeer数に比例する。例えばはてなダイアリーの場合を考えると、1日に1Peerあたり1.31個の記事が書かれるため、それによる1日の通信回数は19万Peerの場合で、1Peerあたり平均約50万回(=1日に自分が書く平均記事数×自分以外のPeer数+1日に他のすべてのPeerが書く記事の合計数の平均×1=1.31×189999+189999×1.31×1)、38万Peerの場合で、1Peerあたり平均約100万回(=1日に自分が書く平均記事数×自分以外のPeer数+1日に他のすべてのPeerが書く記事の合計数の平均×1=1.31×379999+1.31×379999×1)、となり、それぞれ1秒間に5.8回、11.6回という頻度になる。これは多くのサーバにおいて問題となりうる頻度であり、実用上問題であると考えられる。

## 6. 考 察

### 6.1 従来のロボット型検索手法と提案手法との比較

従来のロボット型の検索手法と提案手法を比較する(表1)。ある程度古い文章に対しては従来手法は特に優れているが、クローリング頻度より新しい文章を検索す

ることはできない。提案手法はそういった新しい文章の多くを検索することができる。負荷に関しては従来手法、提案手法ともに実際に検索されるか分からない情報についてもデータを保持する必要がある。検索結果の適合性に関しては提案手法では元々解説されていることが分かっているキーワードについて検索するため、特に優れている。再現性に関してはWeb全体では古い文章の方が多いため従来手法が優れている。

### 6.2 なりすまし、改ざんの防止

各Peerは専用のPKIに基づく公開鍵と秘密鍵の組を生成、保持する。各通知には送信Peerの電子署名を付けることでなりすまし、改ざんを防ぐ。各PeerはPeerリストの中に公開鍵を保持しているので独自に検証することが可能である。

### 6.3 カテゴリとランクの導入

はてなダイアリーキーワードでは、キーワードに「カテゴリ」を導入し、利用者にその言葉の属するカテゴリを明示することで、同音異義語など表記法の違う同じ意味を指す言葉について利用者に選択させる仕組みを用意している。提案手法でも同様にそれぞれのキーワードについてカテゴリを設けることが可能である。また、キーワードは利用者が自由に作成できるため、同じキーワードが複数のPeerで作成されてしまうことがある。利用者により淘汰され解説の充実したページが残るのを待つ方法もあるが、キーワードページに投票フォームを設け、正当性の保証にPKI鍵を用い、利用者によるランク付けを可能とすることで、良質なページを推薦して提示することが可能となる。

### 6.4 応 用 例

ソーシャル・ブックマークサービス(SBS)<sup>10)</sup>への提案モデルの適用を考える。各Peerで、ブックマークしたサイトのURIと内容を表すタグを共有情報とすることで分散SBSが実現できる。また、本稿の適用例で挙げたキーワード共有サービスと組み合わせることにより、キーワードの解説ページと同時に、そのキーワードをタグに持つSBSに登録されたサイトを提示することも可能となる。

## 7. 関連研究

本稿の提案モデルにおけるキーワード共有は、はてなダイアリーキーワード<sup>11)</sup>と見かけ上似ているが、C/Sで完全集中型のそれに対して、提案手法は、分散環境下で、特定のホスティングサーバに依存せず、11)

表1 ロボット型検索手法と提案手法との比較  
Table 1 Comparison between robot type retrieval technique and proposal technique

	古い文章	新しい文章	負荷	適合率	再現率
従来					
提案					

と同様の機能を実現可能である。

P2P 型掲示板の試みは新月<sup>13)</sup>などが挙げられるが、実際のデータ(書き込みテキスト)の複製を伝播させているために、他の Peer でも常に同じ情報が見られるとは限らないといった点で不便である。また、データ発信者の高い匿名性と容易にミラーされる特徴から、隠しファイル埋め込みによるファイル交換の場となる危険性が指摘されている。提案手法では、データの公開元のリンク情報と、キーワード情報を共有する仕組みにより、上述のような危険性が無く、情報流通の利便性の確保が可能である。

ソーシャル・ネットワークサービス(SNS)<sup>14)</sup>の技術的な本質は、利用者同士の親近度に応じたアクセス制御にある。分散型 SNS の試みとして Affelio<sup>15)</sup>などがあげられるが、提案手法は、分散型 SNS と併用可能で、例えば提案手法を適用した Wiki やブログにさらに SNS によるアクセス制御を付与することで容易に SNS 化することが可能である。

16) は、EcAgent 実行環境というサーバを用いた Hybrid P2P で構成されるという点で提案手法と本質的に異なるが、サーバを取り入れれば EcAgent と同様の機能が追加でき、また併用も可能である。

17) は、ブローカレスな P2P ネットワークの構築という点では似ているが、JXTA を基盤とし、Peer としてクライアント PC を想定する 17) が豊富な CPU とメモリによるデータマイニング、嗜好情報・意味情報活用、メタ情報の類似度の活用などの議論に行っているのに対し、提案手法は、低速 CPU、省メモリ、ある程度のストレージ、実時間制約といった制約下で、定期処理を基本とし、即時処理を組み合わせ、エージェント指向の Web 登場の前段階としての Web2.0 の基盤構築を試みている点で異なり、利用ケースが異なる。

Semantic Free Referencing(SFR)<sup>18)</sup>では、分散ハッシュテーブルの探索アルゴリズムを用いた、DNS に依存しない参照解決サービスを提案している。提案手法では P2P ネットワークに参加する全ての Peer の URI をそれぞれが保持することで、P2P ネットワーク上で情報共有を行うことを可能としているが、SFR 上でも提案手法を用い、P2P ネットワークに参加する Peer の SFR 上での識別子のリストを各 Peer が保持することで同様の仕組みが実現可能である。

Ingrid<sup>19)</sup>では、Web サイト毎にインデックスを保持し、複数のサイトにまたがった分散検索を実現する手法を提案している。このような従来の検索がプル型で過去の情報を検索対象とするのに対し、提案手法は未来の登場語に対するプッシュ型検索ということが出来る。つまり「キーワード」を設置することで、未来に対する登場語の逐次観察が可能である。未来の登場語に対しては、その都度全文検索する従来手法より、

登場したサイトから通知を受ける提案手法の方が効率的である。

SIONet<sup>20)</sup>のようなブローカレスな P2P ネットワーク上においても、提案手法のようなプッシュ型の通知機構は有用である。SIONet は意味情報による情報配送機構を持つが、何らかのイベントの発生を基点として発火が行われる。提案手法を SIONet に適用すると、各 Peer に条件を設定してイベントを自動発生させ、逐次観察する機構が実現可能となる。

## 8. 結 論

本稿では、公開サーバ上の CGI を一つの Peer とみなし、キーワード情報を共有することで、情報同士の関係の自律的な構築と潜在的な共通事項の発見が可能で P2P 情報伝播モデルを提案した。また、提案モデルを Wiki に適用し、通常一つの Wiki 内で閉じた自動リンクの関係を拡張し、複数の Wiki 間でキーワードを共有することが可能なシステムを示した。今後、さらに改良を進め、Web2.0 実現のための実用的な基盤技術の一つと位置づけられるよう発展させてゆきたい。

## 参 考 文 献

- 1) Gnutella. <http://www.gnutella.com/>
- 2) JXTA. <http://www.jxta.org/>
- 3) LimeWire. <http://www.limewire.com/>
- 4) Wiki Wiki Web.  
<http://c2.com/cgi/wiki?WikiWikiWeb>
- 5) [http://en.wikipedia.org/wiki/Web\\_2.0](http://en.wikipedia.org/wiki/Web_2.0)
- 6) Perl5. <http://dev.perl.org/perl5/>
- 7) Apache. <http://jakarta.apache.org/>
- 8) PukiWiki. <http://pukiwiki.org>
- 9) tDiary. <http://www.tdiary.org/>
- 10) [http://en.wikipedia.org/wiki/Social\\_bookmarking](http://en.wikipedia.org/wiki/Social_bookmarking)
- 11) はてなダイアリー. <http://d.hatena.ne.jp/>
- 12) Wikipedia. <http://ja.wikipedia.org/wiki/>
- 13) 新月. <http://shingetsu.xrea.jp/index.ja.html>
- 14) [http://en.wikipedia.org/wiki/Social\\_network](http://en.wikipedia.org/wiki/Social_network)
- 15) Affelio. <http://affelio.jp/>
- 16) 中村隆幸, 井上知洋: アドホックコミュニティを発見する EcAgent 実行環境の実装, 情報処理学会論文誌, Vol.45, No.1, pp.94-102 (Jan. 2004).
- 17) 中沢実, 服部進実: 意味情報と嗜好情報に基づく P2P システムの提案と実装, 情報処理学会論文誌, Vol.44, No.3, pp.2666-2677 (Mar. 2003).
- 18) Semantic Free Referencing.  
<http://nms.lcs.mit.edu/projects/sfr/>
- 19) Ingrid. <http://www.ingrid.org/>
- 20) 小柳恵一, 星合隆成, 梅田英和: P2P ネットワーキング技術の提案と紹介, 電子情報通信学会論文誌, Vol.J85-B, No.3, pp.319-332 (Mar. 2002).