コンテンツ間での部分Webグラフ同期方式の提案と特性比較

高砂　幸代† 　小林　亜樹†† 　山岡　克式†† 　酒井　善則††

†,†† 東京工業大学 大学院理工学研究科　〒 152–8552 目黒区大岡山 2–12–1
E-mail: †takasago@net.ss.titech.ac.jp, ††{koba,yamaoka,ys}@ss.titech.ac.jp

あらまし　リンクでつながれたコンテンツ間には意味のつながりがあることが多いことから，Web グラフを利用した様々な研究が過去に行われてきた．しかし，実際に Web グラフを入手するのは困難であった．筆者らが提案してきた協調型 Web アーキテクチャ（CWA）では Web サーバが近傍リンク情報（部分 Web グラフ）の保持，配信機能などを持つ．CWA は単に Web グラフ情報を提供する仕組みだけではなく，リンク切れなどリンクの不整合の防止や、近傍 Web グラフ空間内での検索など，様々な応用が考えられている．CWA の機能を実現するためには，サーバが常に最新の近傍のリンク情報を知っていなければならない．つまり，各サーバは Web グラフが変更するイベント発生時に通知を行い，全ての部分 Web グラフ間で同期を保っている必要がある．本稿では，通知の時間差により発生する，各ノードが保持する部分 Web グラフ情報間の不整合を解消するための同期方式を提案する．具体的には，追加イベントと他のイベントがほぼ同時に発生するこにより，通知範囲が不正になり不整合が生じる．この不整合の原因を明らかにした後，不整合の解消処理を行うノードが異なる，３種類の同期方式を提案する．最後に通知通信量の推計を含む提案方式の特性を比較検討し，実現性を議論する．

キーワード　ネットワーク応用, Web 利用技術, 情報検索, Web グラフ

A Synchronization Method of Web-Subgraph among Contents

Yukiyo TAKASAGO†, Aki KOBAYASHI††, Katsunori YAMAOKA††, and Yoshinori SAKAI††

†,††Tokyo Institute of Technology　2–12–1 O-okayama Meguro-ku Tokyo, Japan, 152–8552
E-mail: †takasago@net.ss.titech.ac.jp, ††{koba,yamaoka,ys}@ss.titech.ac.jp

Abstract　Since hyperlinks connect contents semantically, several studies with Web-Graph have been proposed. However, it was difficult to obtain Web-Graph. In the Cooperative Web Architecture (CWA) we proposed, the Web servers have functions of maintenance and delivery the neighboring link information (Web-Subgraph). CWA is not only a mechanism to obtain the latest Web-graph, but also enables various applications to be expected. For example, “the neighboring search” is a retrieval system using a Web-community, and prevention of the link cutting. To achieve the functions of CWA, the servers have to maintain latest information. That is the servers notify the changes each other when the link information changes, and it is necessary to synchronize all Web-Subgraph maintained in servers. Some changes around the same time poses an inconsistent problem by simple synchronization method. In this paper, we propose the synchronization method to solve this problem among Web-Subgraphs. This problem occurs because of the time difference between notifications. When additional events and other events occur around the same time, the range of notification becomes invalid. Therefore the inconsistent problem occurs. We proposed three synchronization methods, and then we compare and analyze the characteristic of them, and estimate the amount of the notification. Moreover we discuss about possibility from the viewpoint of traffic.

Key words　Applied networking, Technology using Web, Information retrieval, Web-Graph

1. はじめに

現在の HTML では，他のコンテンツへのハイパーリンクを含んでおり，このリンクによりユーザに対するナビゲーションができ，ユーザは関連するコンテンツ間を移動することが可能となっている．インタネットが普及した現在，このリンク情報はコンテンツをノードとし，ハイパーリンクを枝とする大きなグラフ空間（以下，Web グラフ）として考えることができる．リンク情報は意味ある情報として，Web グラフを利用する様々な研究が行われている．代表的なものとしては，被リンク数や，リンク数によりページをランキングする PageRank [1] や，関連する話題は密な Web グラフ構造になっていることを利用して

Web コミュニティを発見する手法 [2]- [4] である．また，検索ツール [5] や可視化ツール [6] にも Web グラフは利用されている．さらに，A. Broder らは Web グラフの構造を解明する研究も行っている [7].

このように，Web グラフは様々な研究分野で利用されているが，簡単に入手することは従来では困難なこととされており，簡単に入手するためにリンク情報を提供するサーバとして LINK データベースが提案されている [8]．しかし，これは Web 上の全てのリンク情報を 1 つのデータベースで一括に保存しているため，頻繁なリンクの変化に対応できず，クローリングを行った後一括で更新を行うため，最新の情報を保持することが出来ない．A.Ntoulas らの調査によると，平均で毎週 25 %の新しいリンクが作られていることがわかっている [9]．このことから，保持する Web グラフの頻繁な更新は必要不可欠だと考えられる．

一方，Web グラフの応用には，被リンクサイトの管理者が，サイト内容の変更や移動などをリンク元に通知したり，リンクでつながれたコンテンツ間には意味のつながりがあることが多いことを利用した近傍 Web グラフ空間内での検索が挙げられる．また，通常の閲覧時においても，正逆両方向の近傍リンク空間の様子を見ながらナビゲーションできれば，目的の情報を探し出したり，自分の閲覧履歴などを一目で理解できる．このように，多くの応用では，比較的近傍の Web グラフが得られればよい．また，Web リンク空間全体の Web グラフを構築しようとするとスケーラビリティの問題に突き当たるが，管理する情報を一部の近傍空間に限定することで，この問題を回避すると同時に，Web の超分散的な管理モデルにも合致させることができる．

そこで筆者らは，最新の Web グラフを Web サーバ側で維持，管理し，利用者や他サーバへ提供する仕組みとして，協調型 Web アーキテクチャ（Cooperative Web Architecture；以下，CWA）を提案している [10] [11]．CWA では，自身の保持するコンテンツの更新を確実に検知できる特性を利用して，Web サーバの付加機能としてサーバ内コンテンツの Web グラフを構築，管理すると同時に，各 Web サーバが近隣の最新の Web グラフ（以下，部分 Web グラフ）を保持することができるよう，近傍コンテンツの更新時に当該コンテンツを保持するサーバからの通知，およびリンク情報の提供を受ける仕組みを導入する．このような付加機能を導入した Web サーバのことを AWS（Advanced Web Server）と呼んでおり，アクセスした利用者に対してコンテンツだけでなく，部分 Web グラフ情報を提供することができる．リンクコンテンツ内での検索機能の提供や，周辺コンテンツを表示する Web ブラウジングなど様々な応用が期待される．

また，従来の Web の問題として，リンク先のページが存在しないといったリンク切れや，アンカーテキストとリンク先のページの内容が異なっているといったリンクの不整合の問題がある．CWA では，リンク情報に変更があった場合，AWS 間で変更を通知しあうことで，リンクの不整合を防ぐこともできる．この通知は，部分 Web グラフを最新のものに保つために必要

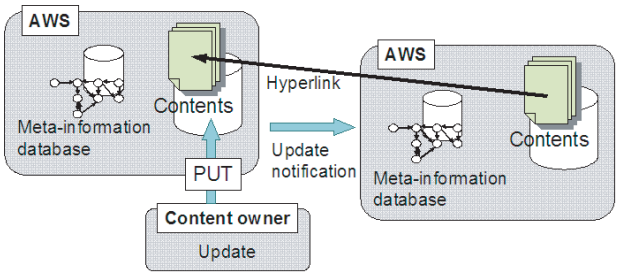

図 1　協調型 Web アーキテクチャ

不可欠なものであるが，単純に変更を通知するのみでは，リンクの変更が同時発生した場合に問題が生じることがあった．そこで本稿では問題の生じない通知方式を提案し，主に通信量の面から比較検討を行う．

2.　協調型 Web アーキテクチャ

CWA では近傍のリンク情報（部分 Web グラフ）とその部分 Web グラフ内のリンクコンテンツのメタ情報をを収集，維持し，それらの情報をユーザに対して提供する．試験的な実装（以下同じ）では，HTTP の GET，HEAD メソッドに対して，当該コンテンツのみでなく，部分 Web グラフやリンクコンテンツのメタ情報も応答する．ユーザは，ブラウザの支援を前提に，そのメタ情報を参考にしてリンクを選択することで Web 上を自在に探索することができる．

また，昨今，いわゆるブログ（WebLog）システムにトラックバックと呼ばれる機能が盛り込まれていることが多いが，これは手動で逆リンクを生成する仕組みであるとも言える．このように，一方的にリンクを設定する行為に対して，逆方向に辿るサービスへの欲求は強いものがあり，HTTP の reference ヘッダによる情報をサーバで収集し，クライアントへ逆リンク情報を提供する手法の提案 [12] などもなされている．CWA では，「リンクによって参照されている」という情報を直接各 Web サーバ間で通知しあい，保持しているため，リンクの双方向での参照が可能であるだけでなく，更新にも追従することができるなど，リンクの機能が強化されている．

さらに，CWA では，近傍検索やナビゲーション支援のための機能のうち，サーバ側に実装したほうが効率的であると考えられる機能についてもサーバ側に付加する．基本的な付加機能として，各 Web サーバが検索キーとメタ情報との距離計算機能を持つ．そのため，各 Web サーバが近傍コンテンツ内での検索を行うことができる．検索処理は，部分 Web グラフとメタ情報を得たクライアント側で行うこともできるし，サーバ側で検索処理を行いその結果のみをクライアントに応答することもできる [13]- [16]．それらの配分は，その時点で実行可能な処理や検索モデルによって定められる．

このような CWA の機能を実現するためには，その最も基本をなす Web グラフ情報について，Web グラフやメタ情報の変化に対応していき，全ての Web サーバが最新の情報を保持し，同期させる必要がある．

これらの情報は，コンテンツの更新時に変化する．コンテン

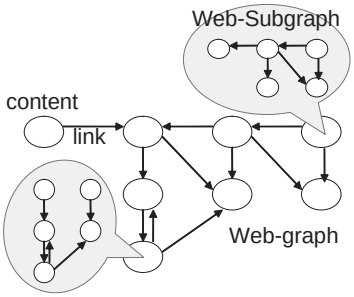

図 2　部分 Web グラフの保持

ツの更新・削除には，例えば，HTTP の PUT・DELETE メソッドを用いることで，Web サーバがコンテンツの更新を確実に検知することができる．Web サーバは更新を検知すると更新されたコンテンツに対して，メタ情報抽出動作を行うと同時に，このコンテンツを含む部分 Web グラフを保持している他の Web サーバへ変更を通知する．通知を受けたサーバはその通知に従って自サーバの保有する部分 Web グラフとメタ情報を更新する．それにより，協調型 Web システム全体でリンク情報やメタ情報の一貫性が保たれ，同期することができる．

しかし，コンテンツの更新がほぼ同時に発生した場合，通知の時間差により保持する部分 Web グラフ情報に不整合が生じることがある．そこで本稿では，この不整合の原因を明らかにし，不整合を解消するための部分 Web グラフ同期方式を提案する．

3.　Web グラフの同期

3.1　問題のモデル化

まず，Web グラフの同期に関する問題をモデル化する．Web コンテンツをノード，ハイパーリンクをリンクとする．ただし，インライン画像などは考えない．このとき，各サーバに多数のノードが格納されることになるが，簡単のため，サーバに関しては後に考えることとし，モデル化では考慮に入れない．

上記から，各ノードが独立に存在する状態を考え，ハイパーリンクを有向リンクで表現すると，ノードから多数のリンクが出力される有向グラフ空間としてモデル化される．ノードには識別のためラベル a,b,... をつけ，リンクを [a→b] のように記述する．

ここで，CWA では別に定める一定範囲（最も単純には，物理的リンク段数で n 段以内，など）の Web グラフ情報を一貫性のある状態に保つことが目的となる．図 2 はノードが 2 段先までの部分 Web グラフを保持している状態を示している．

ここで，あるノードが管理すべき Web グラフの範囲を定める距離的な関数（論理距離）について，物理リンク段数（物理距離）に対する単調性を仮定すると，注目しているノードに別の Web グラフが接続した場合，このノードにおいて保持すべき Web グラフは，もともと保持していた Web グラフと直接接続したノードが保持していた Web グラフの合併集合を上限とすることがわかる．本稿では，簡単のため，この論理距離の物理距離に対する単調性を仮定し，整合性保持という目的のための通知が，議論の Web グラフ内に終始するようにする．

3.2　関 連 方 式

グラフを保有するノード間で同期を保つ方式として，RIP や OSPF といったルーティングプロトコルが考えられる．

RIP ではノードは各ノードまでの最短経路のみを保持している．つまり，ノードへの複数の経路がある場合，１つの経路情報のみしか保持することができない．よって RIP 方式では周辺の全てのリンク情報を保持することができない．

一方 OSPF では，ノードはルーティングドメイン内の全ノードの有効グラフ（LSA）を作成し，保持している．この LSA を常に最新のものにするため，OSPF では一定間隔で HELLO プロトコルとして隣接ノードにメッセージを送ることにより，通信リンクが機能しているか否かをチェックしている．これにより，隣接ノードとの間の通信リンクの削除を感知すると，自身の管理する LSA を変更する．変更された LSA は Flooding によって全ノードに送信される．また，変更がなくても，一定間隔ごとに自身の LSA を全ノードに対して Flooding により送信する．HELLO プロトコルや，Flooding は，ルータ間が物理的に隣接であるために，一定間隔で行ってもトラヒックへの負荷は少ない．しかし，これをコンテンツとリンクに適用すると，コンテンツはリンクにおいて隣接であっても物理的には遠いため非効率となる．

このように，ルーティングの問題と Web グラフ保持の問題は，隣接の物理的な意味が異なるため，ルーティングプロトコルにあるような定期的なチェックは非効率である．

そこで本稿では，変更があった場合のみに通知する方式を採用する．Web グラフの変化するイベントは，リンクの削除・追加，ノードの削除・追加が発生するため，これらのイベント発生時に変化を通知することで，各ノードの保持する部分 Web グラフを同期させる通知方式を提案する．その時の通知範囲は，ノードが保有する部分 Web グラフ内ノードとなる．

4.　単純な同期方式

4.1　単純な通知方式の動作

イベント発生時に変更を通知する単純な通知方式での動作を説明する．ここで，リンクやノードの追加により，新しく隣接となるノードを新隣接ノードとする．また，イベントの発生したノードをイベントノードと呼ぶこととする．

基本的にイベントを通知すればよいが，リンクの追加やノードの追加の追加イベントが発生した場合，新隣接ノードがイベント発生時に保有している部分 Web グラフ情報を知る必要がある．また逆に，新隣接ノードは追加イベントノードがイベント発生時に保有している部分 Web グラフ情報を知る必要がある．そのため，リンクの削除やノードの削除の削除イベントが発生した場合は，イベントを通知するのみで同期を保つことが出来るが，追加イベントの場合はイベント発生時に保有する部分 Web グラフ情報も通知する必要がある．以下に詳しい動作を，イベントごとに説明する．

4.1.1　リンクの削除

リンクの削除イベントノードは，通知範囲の全ノードに，リンクの削除イベントを通知する．

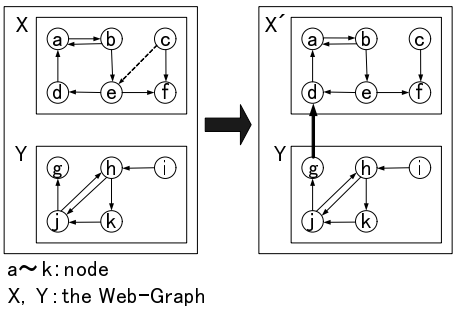

図 3　イベントの同時発生

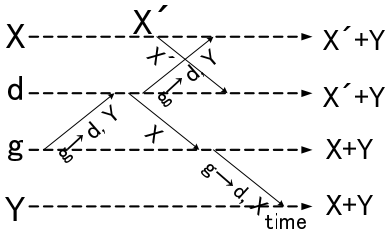

図 4　問　題　点

4.1.2　リンクの追加

リンクの追加イベントノードと新隣接ノードは，互いに保有している部分 Web グラフ情報を交換する．その後，この 2 つのノードから通知範囲の全ノードに，リンクの追加イベントを通知するとともに，交換した部分 Web グラフ情報を提供する．

4.1.3　ノードの削除

ノードの削除イベントノードは，通知範囲内の全ノードに，ノードの削除イベントを通知する．

4.1.4　ノードの追加

ノードの追加イベントノードは，全ての新隣接ノードから部分 Web グラフ情報を入手する．その後，イベントノードは得た部分 Web グラフ情報をまとめ，通知範囲内の全ノードに，自ノードの追加を通知するとともに，入手した全ての部分 Web グラフ情報を提供する．

4.2　単純な通知方式の問題点

単純な通知方式のみでは，各ノードの保有する部分 Web グラフ間に不整合が生じることがある．図 3 のように追加（[g→d] 追加）とほぼ同時に他のイベント（[c→e] 削除）が発生した場合に不整合が生じる．不整合が生じる原因を表しているのが図 4 である．[g→d] 追加により，Web グラフ Y 内のノードには Web グラフ X が通知され，Web グラフ Y 内のノードは Web グラフ X ＋ Y を持つこととなる．しかし，追加通知を受ける前に Web グラフ空間 X で [c→e] 削除が発生し X ´ となっているため，イベントノード c は，イベント発生時に Web グラフ Y 内のノードが通知範囲内のノードであることを知らない．よって，削除イベントは Web グラフ Y 内のノードには通知されず，ノード g や Web グラフ Y 内のノードは古い Web グラフ X を持ち続ける事になり，実際のリンク情報との不整合が生じる．

図 3 ではリンクの追加イベントの場合の問題点を例に挙げたが，ノードの追加の場合も，新隣接ノードから部分 Web グラフ情報を入手してから，追加通知を行うまでに時間差があるため，追加通知前に部分 Web グラフが変更された場合，同様に不整合が生じる．

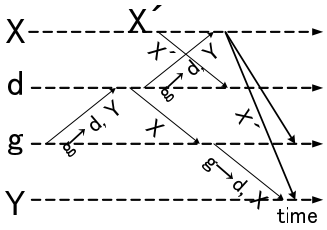

図 5　方　式　A

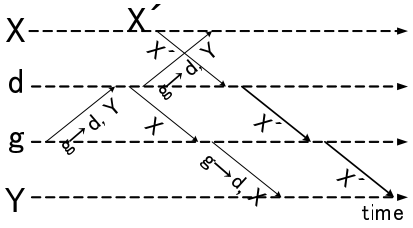

図 6　方　式　B

このように，追加イベントと他のイベントがほぼ同時に発生すると，追加イベントによる新しいリンクノードを通知される前にイベントが発生することにより，拡大された通知範囲にイベント通知を行えないことがあるため，不整合が生じる．

5.　提 案 方 式

4.2 で述べた不整合の問題を解決するための同期方式を提案する．不整合の解消処理を行うノードが，図 3 のような例の場合でいう，追加イベントノード（ノード g），削除イベントノード（ノード c），新しい通知範囲内のノード（Web グラフ Y 内のノード），それぞれである 3 種類の通知方式を提案する．同期方式を設計するにあたっては，Web サーバや途中の通信路が不安定である場合を考慮しつつ，通常の分散型のトランザクションとは異なり，同期を取るべき相手自体が更新する情報に従って増減すると同時に，途中で通知の失敗などがあってもロールバックするべきではない点などを考慮した．

5.1　３方式の動作

5.1.1　方　式　A

方式 A は不整合の解消処理を他のイベントノードで行う方式である．イベントが発生後時間 T の間，イベントノードと新隣接ノードは監視モードに入る．監視モード時に追加ノードの通知を受けたら，それらの追加ノードに対して過去に発生したイベントを通知する（図 5）．

5.1.2　方　式　B

方式 B は不整合の解消処理を追加イベントノードで行う方式である．追加イベントノードと新隣接ノードは Web グラフを交換・提供した後，時間 T の間監視モードに入る．監視モード時に交換・提供した情報に対する変更の通知を受けたら，過去に情報を提供したノードに変更を通知する（図 6）．

5.1.3　方　式　C

方式 C は不整合の解消処理を新しい通知範囲内のノードで行

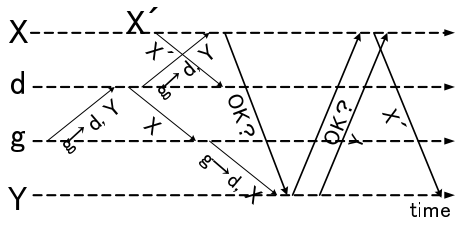

図 7　方　式　C

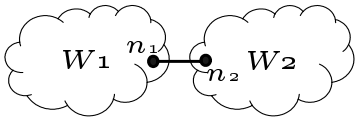

図 8　隣接する部分 Web グラフ

う方式である．追加ノードの通知を受けたノードは情報が正確かを確認するため追加された全てのノードに直接情報を問い合わせる．ここで，直接情報とは自ノードから 1 段先までのリンク情報を指す（図 7）．

5.2　隣接する部分 Web グラフでの伝播

5.1 で提案した 3 方式の正当性を確かめるため，多数のイベントが同時に発生しても同期を保つことが出来るかを考える．4.2 で説明したように，不整合の原因は追加イベントにより，通知範囲が拡大されることに原因がある．そこで，追加イベントにより，部分 Web グラフが連結された場合に整合性が保つことができるかを考えることとする．

まず，追加イベントにより隣接となった部分 Web グラフ間で，変更が正しく伝播されるかを考える．図 8 のように 1 つの追加イベントにより二つの部分 Web グラフ W_1 と W_2 が連結される場合を想定する．このときの部分 Web グラフ間の連結に関わるノードを $n_1(\in W_1)$ と $n_2(\in W_2)$ とする．部分 Web グラフ W_1 内でイベントが発生し，W_1' となった場合に W_2 に W_1' が伝播されるかを考える．

5.2.1　方　式　A

W_1 を W_1' に変化させたイベントに関わる，イベントノードもしくは新隣接ノードは W_1 内に存在する．また，W_2 の情報は必ず W_1 に通知される．よって，W_2 の追加情報を聞いたイベントノードもしくは新隣接ノードは，W_2 に対して W_1' を通知する．以上より，隣接する部分 Web グラフに関する変更は伝播されるといえる．

5.2.2　方　式　B

W_1 内の変更である W_1' は，必ず W_1 内のノードに通知される．よって，n_1 は W_1' を通知され，それを n_2 に通知し，n_2 は W_2 に対して通知を行う．以上より，隣接する部分 Web グラフに関する変更は伝播されるといえる．

5.2.3　方　式　C

W_2 は W_1 の情報を通知されると，W_1 内の全てのノードに対して直接情報を問い合わせる．W_1 を W_1' に変化させたイベントに関わる，イベントノードもしくは新隣接ノードは W_1 内に存在する．よって，W_1' は W_2 に通知される．以上より，隣接する部分 Web グラフに関する変更は伝播されるといえる．

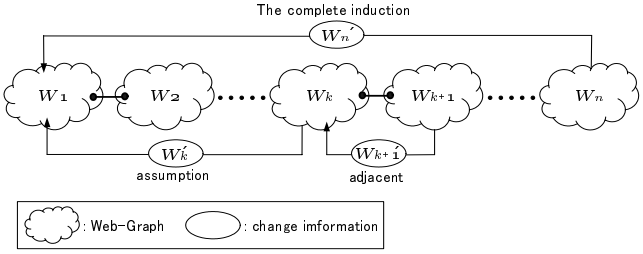

図 9　多連結する部分 Web グラフでの伝播

5.3　多連結する部分 Web グラフでの伝播

次に多数の追加イベントにより，複数の部分 Web グラフが連結される場合について考える．5.2 で説明したように，隣接する部分 Web グラフに関する変更は伝播されることを前提とする．リンクの追加イベントが $k-1$ 個同時発生し，部分 Web グラフ W_1 から W_k が連結した時，W_k の変更は W_1 まで伝播すると仮定する．W_{k+1} が W_k にさらに同時に連結すると，W_{k+1} の変更は隣接である W_k には伝播する．それにより，W_k は W_k'（$= W_{k+1} + W_{k+1}'$）となる．この変更は仮定により W_1 に伝播することになる．つまり，W_{k+1}' は W_1 まで伝播することとなる．以上より，数学的帰納法によって，複数の部分 Web グラフが連結する場合にも，変更が伝播されるといえる．

ここで，問題となるのが方式 A と方式 B における監視モードの時間である．W_N の情報が W_1 に通知されるまでが監視モードの時間内でなければ，変更の伝播は不可能となる．W_N の情報が W_1 に伝播される前に，W_1 は $W_2, W_3, \cdots, W_{N-1}$ と順に情報を得ていく．つまり，監視モードのノードは，追加イベントに伴う処理を近い場所で起こった追加イベントから順に行っている．そこで，監視モードのノードは，処理を行っている間は監視モードを延長することで，W_N の情報が W_1 に通知されるまで監視モードであり続けることができ，複数の部分 Web グラフが連結する場合にも，変更が伝播することが可能となる．

このように，追加イベントが複数同時発生した場合も，Web グラフの変更を正しく伝播し，整合性を保つことができる．つまり，どのようなイベントが同時に発生しても整合性は保たれるため，提案方式は有効であるといえる．

6.　方式特性比較と実現性

6.1　3 方式の特性比較

3 方式の通知量の比較を行う．表 1 は方式ごとの各イベントでの通知量をまとめたものである．l_k は k 段先のノード数の平均値とし，l_0 は 1 とする．方式 C では，追加イベントで直接情報を聞きにいくため，式 (1)(2) の通知量が方式 A，B の通知量に追加される．

$$\text{リンクの追加}：4\sum_{j=1}^{n-1}\sum_{k=0}^{n-j} l_{j-1}l_k \tag{1}$$

$$\text{ノードの追加}：2l_1^2\sum_{j=1}^{n-1}\sum_{k=0}^{n-j} l_{j-1}l_k \tag{2}$$

方式 C は常に直接情報を確認するため，方式 A と B に比べ

表 1　3 方式の通知量

	方式 A&B	方式 C
リンクの削除	$\sum_{k=1}^{n} l_k$	$\sum_{k=1}^{n} l_k$
リンクの追加	$\sum_{k=1}^{n} l_k$	$\sum_{k=1}^{n} l_k + 4\sum_{j=1}^{n-1}\sum_{k=0}^{n-j} l_{j-1}l_k$
ノードの削除	$\sum_{k=1}^{n} l_k$	$\sum_{k=1}^{n} l_k$
ノードの追加	$\sum_{k=1}^{n} l_k$	$\sum_{k=1}^{n} l_k + 2l_1^2\sum_{j=1}^{n-1}\sum_{k=0}^{n-j} l_{j-1}l_k$

確実性は高いが，通知量が非常に大きくなる．それに対し，方式 A と B は監視モード時間内の同時発生のみしか保証が出来ないため，確実性は劣るが，無駄な通知量が少ない．

方式 A と方式 B は通知量は等しいが，監視モードに入るノードの数に相違がある．方式 A ではイベントが発生したノードが全て監視モードに入る．そのため監視モードに入るノード数が多いが，イベントが同時発生した場合の通知量が分散されるというメリットがある．それに対し方式 B では追加イベントが発生した場合のみにノードが監視モードに入る．そのため監視モードに入るノード数が少ないというメリットはあるが，イベントが同時発生した場合の通知量が少数のノードに集中する．これらの検討により，方式 C は通知量が大きいため現実的な方式ではないと考えられる．方式 A と方式 B については実装上どちらが優れているか今後の検討課題である．

6.2　通知の集約と実現性

CWA では，サーバがサーバ内のコンテンツの部分 Web グラフを一括して管理するため，実際の通知は同サーバ内のコンテンツに対しては行う必要がなく，他サーバ内のコンテンツのみに一括して行えばよい．よって，実現性に関する議論は通知をサーバ単位で集約することを考慮して行う．

n 段先までのリンクコンテンツを保有しているサーバ数を S[個]，サーバの一ヶ月の更新数を u[回/月] とし，サーバの一ヶ月のアクセス数を A_s[回] とすると，通知を行うことで増えるアクセス数の割合 P は

$$P = \frac{S \cdot u}{A_s} \tag{3}$$

となる．

Web では 1 つのコンテンツは，例えば 1 つの HTML 文書とその部品を構成する複数の HTML 文書や画像ファイルというように，複数のオブジェクトから成り立っている．よってユーザがコンテンツを取得するためには，複数の HTTP 要求メッセージと HTTP 応答メッセージの通信が行われる．したがって，式 6.2 の分母にあたるサーバへのアクセス数 A_s は，単なる HTML ページへの閲覧数よりもはるかに多い．また，他サーバ内のリンクコンテンツを多く保有するコンテンツは，現在でも他サーバからのリンクを辿って多くのアクセスがあり，コンテンツの知名度が高いと考えることができる．つまり，S が大きいコンテンツは A_s も大きいと考えることができる．よって部分 Web グラフを保有する範囲を適切に選択することで，爆発的にトラヒック量が増えるような深刻な状況はないと考えられるが，定量的評価は今後の課題である．

7. ま と め

本稿では，様々な応用が期待されている CWA を実現するために必要不可欠な，部分 Web グラフ間での同期を保つための通知方式の問題点を明らかにし，不整合を解消する通知方式を提案した．また，通知通信量の推計を含む特性の比較検討を行い，実現性を議論した．今後は，リンクコンテンツを保有しているサーバの平均数を調べることで，現在のアクセス数に対する増加トラヒックの割合を考慮し，実現可能な部分 Web グラフの保有範囲について検討を進めていく．また，提案した方式を実装により評価し，３つ提案方式を実装の面から比較していく．

文　献

[1] L. Page: “PageRank: Bringing order to the Web”, Stanford Digital Digital Libraries working paper, 1997.

[2] Jeffrey Dean, and Monika R. Henzinger: “Finding related pages in the World Wide Web”, In Proceedings of the 8th WWW Conference, 1999.

[3] R.Kumar, P.Raghavan, S.Rajagopalan, and A.Tomkins: “Trawling the web for emerging cyber-communities”, In Proceedings 8th WWW Conference, 1999.

[4] 豊田正史: “WWW における関連コミュニティ群の発見”, 情報処理学会研究報告 DBS122-40, IPSJ, 2000.

[5] 久我昌崇, 中所武司: “Web コミュニティの知識に基づく情報検索手法の評価”, 情報処理学会研究報告 DBS123-6, IPSJ, 2001.

[6] Nahum Gershon: “Moving happily through the world wide web”, In *IEEE Computer Graphics and Applications*, pp. 72–75, 1996.

[7] Andrei Broder, Ravi Kumar, and Marzin Maghoul: “Graph structure in the web”, In Proceedings 9th WWW Conference, 2000.

[8] K.Randall, R.Stata, R.Wickremesinghe, and J.Wiener: “The LINK database: Fast access to graphs of the Web”, Research Report 175, Compaq Systems Research Center, 2001

[9] A.Ntoulas, J.Cho, and C.Olston: “What’s New on the Web? The Evolution of the Web from a Search Engine Perspective”, In Proceedings 13th WWW Conference, 2004.

[10] Aki Kobayashi, Katsunori Yamaoka, Yoshinori Sakai: “Cooperative Web Architecture for Search and Navigation Ass istance”, Proc. of ICWI2002 IADIS WWW/Internt 2002, pp.726-729, 2002.

[11] Aki Kobayashi, Kuangmin Tan, Katsunori Yamaoka, Yoshinori Sakai: “Relevant information retrieval for cooperative Web architecture”, Proc. of ICWI2004 IADIS WWW/Internet 2004, pp.1125-1128, 2004.

[12] Soumen Chakrabarti, David A. Gibson, Kevin S. McCurley: “Surfing the Web Backwards”, In Proceedings 8th WWW Conference, pp.1679-1693, 1999. Toronto, Canada Pages: 1679 - 1693 Year of Publication: 1999

[13] 小林 亜樹, 新名 崇, 内藤 清一郎, 酒井 善則, 山岡 克式：“PIRCS: リンク情報の自律的収集による Web 情報探索”, 電子情報通信学会技術報告, SSE2000-116, pp.7–12, 2000.

[14] 樋山 大輔, 内藤 清一郎, 小林 亜樹, 山岡 克式, 酒井 善則：“超分散型 Web 検索システム PIRCS の試作”, 電子情報通信学会技術報告, SSE2000-239 IN2000-195, pp.25–32, 2001.

[15] Aki Kobayashi, Hidetomo Miyahara, Kaito Tochihara, Hengjiang Wang, Katsunori Yamaoka, Yoshinori Sakai : “PIRCS: A Link Context Based Search on The Web,” Proc. of IEEE PACRIM’03 S11-1, pp514-517, 2003.

[16] Hidetomo Miyahara, Aki Kobayashi, Katsunori Yamaoka, Yoshinori Sakai: “Visualization of Web Link Space for Neighboring Search”, Proc. of ICWI2004 IADIS WWW/Internet 2004, pp.1135-1138, 2004.