5K-7

大規模 Web アーカイブにおける時間分解能向上手法の検討

田村 孝之† 喜連川 優‡

三菱電機 情報技術総合研究所† 東京大学 生産技術研究所‡

1. はじめに

筆者らは社会分析を目的とした Web 時空間解析に取り組んでおり，その基盤として日本の Web 情報を網羅的に収集・蓄積した大規模な Web アーカイブの構築を進めて来た[1]．Web 空間全体を一括収集する従来型クローラでは 1 週間程度の時間分解能が下限となるため，個々の Web ページを独立した周期で繰り返しアクセスする連続的クローラを開発し，時間分解能 1 日を実現している[2]．しかし，ニュースサイトやブログの中には 1 日に複数回更新されるものも多く，より高い時間分解能の実現が求められる．特に，最近のミニブログの流行により，twitter の public timeline[3]等，リアルタイム的に更新される URL も一般的になりつつある．

連続的クローラは，再アクセスの周期を最適化するためのアクセス履歴のサマリや，Web サーバへの HTTP 要求を構成するためのパラメータを Web ページ毎の状態情報として管理しており，10 億ページを超える大規模クローリングではその容量は 1TB 規模に達する．そのため，単純にアーカイブに追加されるコンテンツ本体よりも，Read-Modify-Write サイクルを要する状態情報の方がアクセスコストが高く，時間分解能向上のボトルネックとなる傾向にある．本稿では，状態情報の更新処理を高速化するためのデータ構造を紹介するとともに，Web アーカイブの時間分解能向上に対する課題と対策について議論する．

2. Web アーカイブ更新処理

既知 URL の再収集による大規模 Web アーカイブの更新処理の概要を図 1に示す．時刻 *t* において，クローラは二次記憶上の状態情報 DB からアクセス対象の URL とその属性（*attr*, …）を取得し，対応する Web サーバに HTTP 要求を送付する．次いで HTTP 応答としてコンテンツを取得し，アーカイブへの追記を行う．さらに，URL の属性のうちアクセス履歴のサマリや HTTP パラメータ（最終更新時刻（Last-modified），タグ文字列（Etag）等）を更新して新たな属性値（*attr'*, …）を算出するとともに，属性中の再アクセス周期に基づいて次回アクセス時刻 *t'*を決定し，状態情報 DB の更新を行う．

ここで，状態情報 DB は，URL を主キーとする通常の検索構造ではなく，次回アクセス時刻に基づくキュー構造として実現している．すなわち，次回アクセス時刻を有限の時間分解能で表現し，同一の次回アクセス時刻を持つ状態情報レコードをまとめて二次記憶上の同一領域（以下，バケットと呼ぶ）に格納する．ただし，現在時刻 *t* と次回アクセス時刻 *t'*の差 *t'*- *t* には上限を設け，バケット数は有限（*B* 個以下）とする．

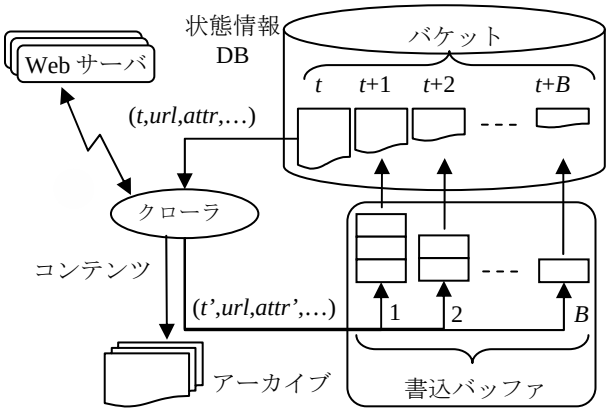

図 1 Web アーカイブ更新処理の概要

クローラは，現在時刻 *t* に対応するバケット *t* を先頭から読み込み，含まれる状態情報レコードに基づいてアクセス対象 URL を受動的に決定する．これにより URL をキーとして状態情報レコードを検索する必要がなくなり，二次記憶からの読み込みをシーケンシャルアクセスとして実現することが可能になる．

一方，次回アクセス時刻 *t'*はレコード毎に異なることから，更新後のレコードは一旦書込バッファに格納される．書込バッファは *t'*- *t* に対応して存在し，固定長のページ単位で拡張される．全ての書込バッファのページ数の合計が一定値を超えた場合，各バッファを *t'*- *t* の昇順に処理し，対応するバケットの末尾にレコードを追記してページの解放を行う．この動作により，二次記憶への書き込みのほとんどをシーケンシャルアクセスとすることが可能になる．

クローラは，バケット *t* のレコードを全て処理し終えると当該バケットを削除し，その領域を再利用可能にする．続く時刻 *t*+1 においては書込バッファ 1 のレコードをバケット *t*+1 に追記し，その記憶領域を解放する．次に書込バッファ *i* (2≦*i*≦*B*) をそれぞれ書込バッファ *i*-1 に読み替え，書込バッファ *B* とバケット *t*+*B*+1 を新たに作成する．続いて，バケット *t*+1 を対象として上述の処理を繰り返す．

On Improving Time-resolution of a Large-scale Web Archive.
†Takayuki Tamura, IT R&D Center, Mitsubishi Electric Corp.
‡Masaru Kitsuregawa, IIS, The University of Tokyo

3. Web アーカイブ時間分解能の更新処理性能への影響

実際の連続的クローリングにおいて，Web ページの更新傾向から推定した更新間隔の頻度分布を図 3に示す．ここでは，64 ノードによる並列クローリングにおいて 1 ノードに割り当てられた URL のうち，2009 年 2 月時点で 1 回以上正常にアクセスできた URL 17,496,056 件を用いた．アクセス周期の時間分解能は 1 日とし，その上限を 1 年とした．HTTP 応答の最終更新時刻（Last-modified）を利用しているため，最小推定値は 1 日より小さく，約 30 分となった．

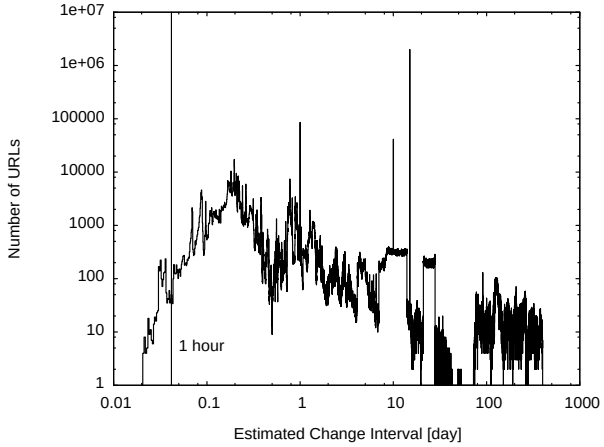

図 3　Web ページ更新間隔推定値の頻度分布の例

表 1　単位時間当たりアクセス URL 数

時間単位	単位時間当たりアクセス URL 数	
1 分	5.13K	(0.029%)
1 時間	0.346M	(1.98%)
1 日	2.83M	(16.2%)
1 か月	11.4M	(64.9%)
1 年	17.5M	(100%)

異なる時間分解能に対し，更新間隔推定値をアクセス周期とする連続的クローリングを行った際の，単位時間当たりアクセス URL 数を表 1に示す．括弧内は全ページに対する割合である．時間分解能を向上させるにつれ，アクセス対象の URL が急激に減少することが確認できる．前述の状態情報 DB においては，小さなバケットが多数作成されることになり，シーケンシャルアクセスの効果が減少すると予想される．

異なる時間分解能に対する状態情報 DB 更新処理コストの測定結果を図 2に示す．ここでは，コンテンツの取得は行わず，図 3の 17,496,056 URL に対応する状態情報 DB の各レコードを，その内容に従って周期的に更新し続けた際の 1 レコード当たり処理時間のみを測定した．時間分解能によりバケット数や各バケットのサイズが変化するが，状態情報 DB の総容量は約 2.7GB であり，その変化は

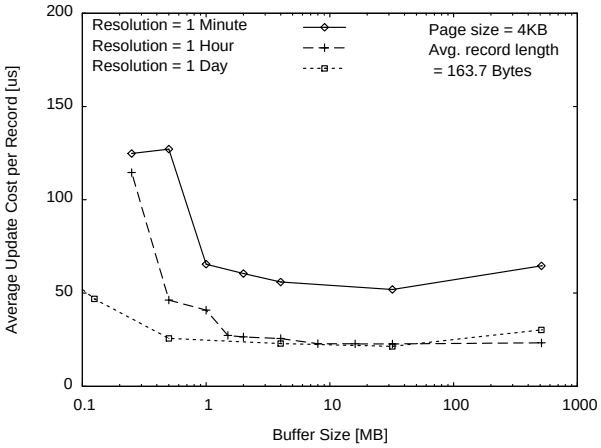

図 2　書込バッファ容量に対するレコード当たり更新コストの変化

2%程度に止まった．

時間分解能が 1 日の場合，更新コストは書込バッファ容量に関わらず安定している．これに対し，時間分解能を 1 時間とした場合は，バッファ容量が 1MB を下回ると更新コストが急激に増大している．さらに時間分解能を 1 分とした場合は，バッファ容量が十分であっても更新コストが全体的に高くなっている．表 1から予想される通り，1 分程度の時間分解能においては高い効率を維持することが難しいことが確認できた．この問題に対する対策として，アクセス間隔と時間分解能の単純な積に対応する数のバッファを用意する代わりに，大きなアクセス間隔に対する時間分解能を大きく（粗く）し，バッファ数を削減することが考えられる．ただし，同一バケット内に異なるアクセス時刻のレコードが混在するため，その再スケジュール処理が必要となる．

4. おわりに

大規模 Web アーカイブの時間分解能向上に伴う課題を実際のデータに基づいて示し，その対策について検討した．今後は実装を行い，その効果の検証を行う予定である．

謝辞

本研究の一部は文部科学省委託事業「多メディア Web 解析基盤の構築及び社会分析ソフトウェアの開発」による．

参考文献

[1] 喜連川, 豊田, 田村 他：“Socio Sense：過去 9 年に及ぶ Web アーカイブから社会の動きを読む”, 情報処理, 49(11), pp. 1290-1296 (2008)
[2] 田村, 喜連川：“大規模 Web アーカイブのための更新クローラの設計と実装”, 日本データベース学会 Letters, 6(1), pp. 173-176 (2007).
[3] Twitter, Inc.: “Twitter API Documentation”, http://apiwiki.twitter.com/Twitter-API-Documentation, accessed 2009/12/24.