D-11　利用状況を考慮したWebキャッシュの一致性維持方式

Cache Consistency Policy Based on Access Patterns of Web Users

王　波†　　成　凱†　　上林　弥彦†
Bo Wang　　Kai Cheng　　Yahiko Kambayashi

1. まえがき

インターネットやWebの急速な普及に伴いネットワークトラフィックは年々倍増しつつあり、Web利用の効率性は利用者にとって一番気になる問題であるとGUVによる調査で分かった。この効率を向上するためにキャッシュがよく使われている。どのデータをキャッシュに入れるかは利用状況で優先度を決め利用の少ないデータはできる限りキャッシュに入れない。しかし、問題は元のデータが更新された場合にはデータがキャッシュに入っていても利用できないことになる。キャッシュされたデータは利用できるかどうかを判明するために元のデータと一致しているかどうかチェックしないといけない。

キャッシュデータの一致性を維持するには更新状況（例えば、更新頻度）のみを考えた方式はもっとも一般的であるが、Webキャッシュの場合はデータの量が非常に多いので、更新ごとに一致性をチェックするのはシステムに大きな負担になりうる。本論文では、更新状況だけでなくWebデータの利用状況などの情報も活かして効率的な一致性維持方式を提案した。この方式の基本的な考えは次のようである。

- 利用が多く更新が少ない場合は更新頻度に従って一致性チェックを行う
- 利用が少なく更新が多い場合は利用ごとに一致性チェックを行う
- もっと利用が少ない場合は、一致性をチェックせずに古いデータでもよい

実際には利用度や更新度が時間的に変化するので更新頻度・利用頻度も動的に再計算する必要がある。この方式により、再利用データの新鮮度を維持できるとともに、一致性を維持するためにネットワークやシステムにかける負担も軽減できた。この方式の詳しいことについて以下に述べる。

2. 更新状況に基づく一致性維持方式

データの新鮮度への要求の厳しさによってキャッシュ一致性は強い一致性（古いデータを絶対にクライアントに返さない）と弱い一致性（一部古いデータを返しても構わない）がある。また、一致性チェックのタイミングによって、能動的（要求中のデータ以外のデータもチェック）一致性チェックと受動的（要求されているデータのみチェック）一致性チェックがある。一致性を維持するため、キャッシュされたデータは古くなったかどうかについて調べることが重要である。現在主に次のような方法がある[2]。

1）クライアントポーリング（Client Polling）法：クライアント側で、キャッシュされたファイルと元のファイルのタイムスタンプを比べ、新たな更新のあるデータを廃棄し新しいバージョンをダウンロードする。2）サーバインバリデーション（Server Invalidation）法：クライアントの代わり、サーバ側ではデータの更新のある時にこのデータをキャッシュしているプロキシサーバに知らせる。この方式は無駄なネットワーク使用の節約ができるが、サーバがプロキシサーバのキャッシュ状況を把握しないといけないため、スケーラビリティやセキュリティやプライバシーの問題がある。

3）TTL (time to live)法：HTTPヘッダーの一種として、データの有効期限を用いる方法である。この期限を切れた場合は、キャッシュデータを捨てられ新しいバージョンを取り入れる。4）IMS (If-Modified-Since) 法：HTTPのキャッシュ関連のメソッドであり、元のデータが更新のあった場合のみ、新しいバージョンをダウンロードする。

ここで極端のサーバ側の協力を得られる場合と毎回ポーリングの場合を除くと、その以外の場合はデータの更新状況のみに応じて一致性チェックを行っている。

3. 利用状況を考慮した一致性維持方式

本章では、能動的に利用状況に応じる一致性維持方式を提案する。この方式では以下のような要素を考える。

Webキャッシュで更新を扱う

1. 利用者データの利用。キャッシュに本来用いられる利用情報である利用の最近性によって、最近あまり利用されていないデータに対して一致性チェックをしなくてもよい。最近よく利用されたデータは再利用の確率が高いので、優先的にチェックする必要がある。
2. 更新頻度に対する過去の情報の利用。これまでの更新状況に基づいて更新頻度を計算しデータの有効期限（TTL）を決定する。更新の頻繁なデータを小さいTTL、更新の少ないのは大きいTTLを与える。
3. 利用重要度と組み合わせ、重要でないデータは更新されて若干古くなっても一致性チェックをしなくてもよい。
4. システムの利用度。システム負荷は重い時、大事なデータでなければチェックしなくてもよい。しかし、システムの利用が少ない時は能動的に多くのデータの一致性チェックを行う。
5. 更新があった場合は利用者に知らせる（push）。更新があった場合でも、利用者に知らせて新しいバージョンをダウンロードするかどうかの選択は利用者によって決まる。

以上の要素を考えて一致性チェックの優先度が決められる。まず、riはデータiの前回アクセス以来の時間、fiはiの利用頻度(最近1週間の一日平均利用回数)、uiはiの更新頻度に基づく有効期限TTL（日数、ui ≥1）、wiはiの

†京都大学大学院情報学研究科社会情報学専攻

利用重要度とする。データ i の一致性チェックの優先順位は次のように決める。

$$P(i)=(w_i \cdot f_i)/(r_i \cdot u_i) \qquad (1)$$

数式（1）によると、最近利用されたデータ、よく利用されたデータ、及び頻繁に更新されたデータは優先的にチェックする。さらに、重要度の高いデータも優先的に同期させる。この時刻で利用するデータ iに対しては $r_i \to 0$ なので p(i)は最も大きいとなるので、最も優先的にチェックするわけであるが、実はそうでもない場合がある。

まず、もしシステム負荷が高ければ、重要でないデータは更新しなくてもよい。次に、実際に利用が多く更新が少ない場合は更新頻度に従って一致性チェックを行う。利用が少なく更新が多い場合は利用ごとに一致性チェックを行う。もっと利用が少ない場合は、一致性をチェックせずに古いデータでもよい。

4. システム設計

提案方式を検証するために我々は次のようなセマンティック Web キャッシュシステムの実装を行っている[1]。ここで、Webデータを HTTP オブジェクト、物理ページ、論理ページ、セマンティック領域の四つの階層に抽象し、それぞれのキャッシュ優先キューによって管理される（図1）。例えば、ある HTML ファイルと複数のメディアファイルを HTTP オブジェクトとして、一つの物理ページを構成する。複数の関連深い物理ページを一つの論理ページにまとめ、サッカー選手というセマンティック領域に属する。

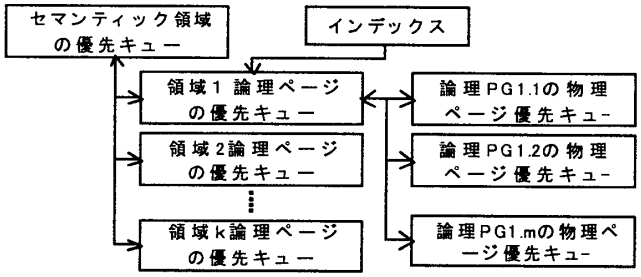

図1：複数の優先キューによるキャッシュ管理システム

Web データはこのような方式で管理すると、セマンティック領域の重要度から個々の HTTP オブジェクトの重要度も得られる。また、キャッシュ管理に本来必要とする利用状況データは更新優先度の計算に直接応用できるので、別に手間がかからない。

更新頻度の計算には次のようなλ-Aging法を用いる。

新しい平均更新頻度＝λ・最近の更新頻度＋
(1-λ)・これまでの平均更新頻度

$0 \le \lambda \le 1$。λが大きければ大きいほど、最近の更新頻度を重視し、前の状況をより速く「忘れて」ゆく効果がある。このようにすれば、更新履歴を記録する余分のメモリスペースが要らなくなる。

重要な更新を利用者に知らせる機能は[1]に述べた SVL（共有 VisitedLinks）機能と統合すればよいが、利用者が不在の場合に対応するため、メールなどプッシュ型の情報伝達手段を用いる必要がある。

5. ISP の利用データを用いた予備実験

キャッシュ手法の検証にトレースベース実験がよく採用されているが、ほとんどの HTTP サーバやプロキシサーバのアクセスログにデータ更新の記録が残されていない。このため、ログデータに基づいて、更新の状況を人工的に加入することが必要である。

我々は京都市の ISP I-NET から一ヶ月と一週間のプロキシサーバのアクセスログをそれぞれ入手できている。この一週間のデータは次のよう内容がある。

Proxy 数	時間	データ量	Request 数	Client 数
26	7日	1.07GB(.gz)	8千4百万	2万2千

このアクセスログを解析して分かるように利用されたデータは 60%以上がイメージデータであり更新は少ないと考えられる。TEXT/HTMLデータは 30%近くを占めている。このうち、再利用されたことのあるデータは 35%しかなく、実験はこの 10%前後のデータの更新状況と利用状況に絞る。また、利用された実のデータが入手できないので重要度の考慮も今回の実験から省くことにした。

一致性維持手法の効果を測るには、EA (Effective Average Age)を用いる。

$$EA=\sum_i \sum_t (Age(i,t) \cdot y_{i,t})/N,$$

N はこれまで見たリクエストの総数、$y_{i,t}$は時刻第 t 番目のリクエストの時（時刻 t）データ i がアクセスされたか(1)されていないか（0）の指示変数である。Age(i,t)は時刻 t の時にデータ i の古さである。これは実データに含まれていないので、人工的に合成する。

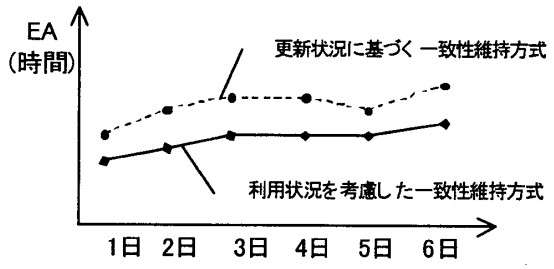

図2 予備実験の結果

6. まとめ

本論文はデータの利用状況を考慮した新しいキャッシュ一致性維持方式を提案し実験検証を行った。有効平均老化指標（EA）の結果は提案方式の有効性を証明した。

参考文献

[1] 成 凱，上林 彌彦，Web データのセマンティックキャッシュ手法，2002 夏のデータベースワークショップ (DBWS2002) 2002-DBS-128. pp. 383 – 389

[2] Greg Barish et al. World Wide Web Caching: Trends and Techniques, *IEEE Communications Magazine, Internet Technology Series*. pp.178-185