モーバイルエージェントを使ったスケーラブルなWWWサービス

Mobile Agents based Scalable WWW service

岸本 克巳[†] 谷口 祐治[††] 河野 真治[†] 玉城 史朗[†]
Katsumi KISHIMOTO Yuji TANIGUCHI Shinji KONO Shiro TAMAKI

†琉球大学情報工学科
Department of Information Engineering, University of the Ryukyus
††琉球大学総合情報処理センター
Center for Integrated Information Processings, University of the Ryukyus

概 要

WWWトラフィックの削減には主にプロキシサーバを用いるが，単純なキャッシュでは重複したWWWリクエストを削減することができてもキャッシュのヒット率が低いので効果が少ない．この問題を解決するために，これまでにリモートサーバで先読み処理を行うエージェントを実装した．次に新しい手法として教師信号を採用した先読みエージェントを実装し，評価を行った．

これらを踏まえ，WWWブラウザで収集される利用パターンに応じて定期的なWWWプロキシ間の配信を行う手法と，モーバイル・エージェントを使ったWWWサーバとWWWプロキシ，WWWブラウザを結ぶプロトコルを提案する．ここではJavaの移動エージェントであるAglets [1] [2]を用いて実装と評価を行った．

1 はじめに

Internetでは代理サーバに貯め込んだキャッシュを使ってクライアントの要求に答えるプロキシを用い，通信量自体を減らす様々な研究と評価が行われている．

これまでの単純なキャッシュではクライアントに要求された資源しかキャッシュしないため，サーバの規模を大きくしてもヒット率は上がらずネットワークトラフィックの減少に効果がない．ヒット率をp，$T_{hit} = 5$秒，$T_{miss} = 60$秒とした場合，$T_{avg} < 10$秒に要求されるヒット率を次式で計算すると約91%となる．しかし実際には30%を超えるのは困難である．

$$T_{avg} = T_{hit} \times p + T_{miss} \times (1 - p)$$

そこでプロキシにリンクの先読みを実装することでヒット率の向上が報告されている[3]．しかし，e_1，e_2の2つのHTML文書に各々$\{f_1, f_2, f_3\}$，$\{f_2, f_3, f_4\}$の画像ファイルが含まれていた場合には重複した取得要求が発生することから，従来の先読みでは参照されないキャッシュや，要求の重複によるネットワークトラフィックの増加が問題になる．

これらはクライアントから要求されたURLの資源のみをクライアントへ渡すために起こっている問題であり，問題解決にはWWWサービスにインテリジェンスを導入する必要がある．

2　エージェントによる先読み

WWWサービス高速化の簡単な方法に，参照されるHTML文書に含まれるリンクの先読みがある．Aglets移動エージェントを用いてプロキシサーバに要求されるURLを取得し，リモートサーバでキャッシュの先読み処理を行うプログラムを作成した．

2.1　エージェント生成のプロセス

先読みエージェントはObserver　Agletと Prefetch Agentで構成される．先読み処理はサーバに大きな負荷をかけるため [3]，エージェントはリモートサーバへ移動して処理を行う．エージェントによるサーバ間通信は以下の手順で行われる (図 1)．

1. Observer AgletはPrefetch Agletの派遣先サーバを探索する，
2. 派遣先サーバが見つかると，Observer AgletはPrefetch Agletを派遣する，
3. 派遣されたPrefetch Agletは派遣先のサーバで処理を行う．

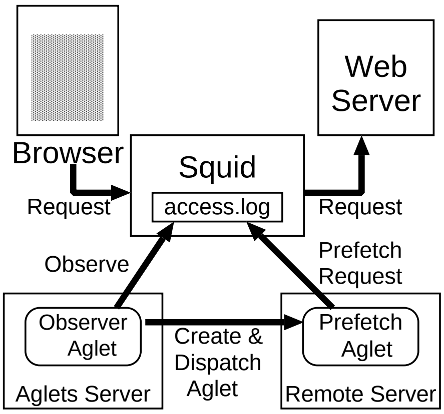

図1　先読みエージェントによるプロキシサーバ構成

2.2　エージェントの基本方針

エージェントの基本的な動作を決めるために，Squidによるプロキシサーバの運用を行った [4] [5] [6]．この結果から直接キャッシュを取得する時間は平均4.67秒となった．また，WWWでは自明な参照局所性が無いことから，以下の基本方針を決定し実装を行った．

1. Squidに要求されたリクエストが，蓄えられたキャッシュにヒットしなかった場合にのみAgletsサーバ上で子エージェントが生成される．
2. エージェントの生成は1つの要求に対して1つ生成される(1クライアントごとではない)．
3. 子エージェントは先読み処理を開始して20秒以内に処理が終らなければ処理を途中で中断し，消滅する．
4. 先読みするリンクが20を越えた場合は先読み処理を行わない．

2.3　運用実験

先読みエージェントの試験運用を琉球大学キャンパスネットワークで行った．プロキシ網はseagull(Memory:256MB,Disk:8GB)，capoeira(Memory:128MB,Disk:1GB)の2台で構成し (図2)，先読みしたデータはseagullにキャッシュした．子エージェントの派遣先にはseagull，capoeiraを含め計5台のサーバを利用した．

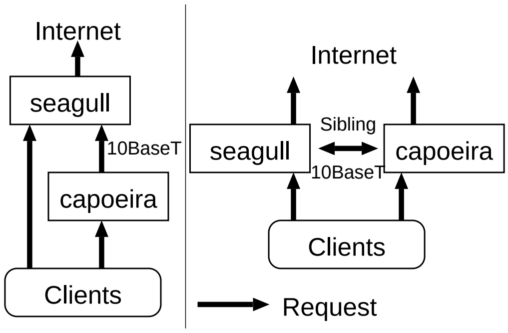

図2　階層構成 (左) と並列構成 (右)

2.4　実験結果

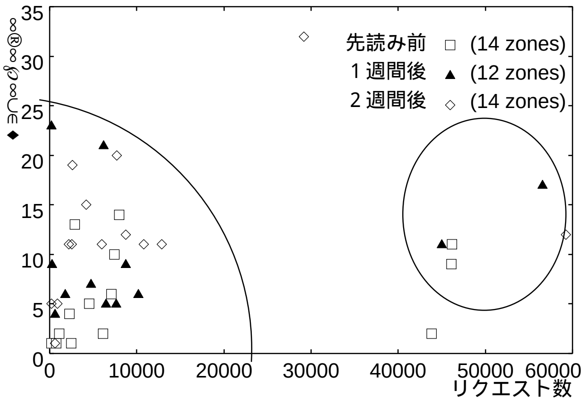

図3　階層構成の実験結果

図3の左側の円はseagullに直接プロキシ接続しているクライアント、右側の円はcapoeiraや利用の多いゾーンの分布を示している．実験結果は利用者数が少ないため、1ゾーンあたり約10クライアントにまとめて表示した．

この実験結果より，単純なキャッシュや先読みはネットワーク形態や利用状況に応じたキャッシュが行われていないため，ネットワークトラフィックを増加させることになった．

図3の結果では先読みを行うことでヒット率が底上げされている．しかしseagullがキャッシュした総データ量の3340.0Mbytesのうち先読みのデータの占める割合は2845.6Mbytesとなり，クライアントごとの平均ヒット率は7%であった．並列構成の場合も同様の結果が得られた．これより単純な先読みは逆にネットワークトラフィックを増加させる結果となった．

3 教師信号付きエージェント

前節の実験結果から本研究では先読みに加え，教師信号を採用した先読みエージェントを作成した．本稿での教師信号とは，特定集団の行動を決定，又は次の行動が予測できる情報と定義する．例えばインターネット講習会では講師があるURLを指定してデモンストレーションを行い，生徒に練習をさせると生徒は講師の真似をして同じURLを指定し，同じホームページを見たりする．この場合は教師が指定したURLが教師信号になる．

ここでは教師信号を用いてある目的集団の行動を把握し，そこに先読みを行う手法を用いて実験を行った．この方法によって集団や個人の利用パターンの取得とヒット率の向上が確認できれば，様々な利用パターンに対応したWWWサービスの実現が期待できる．教師信号には講師端末からのhttpリクエストを採用した．

3.1 システム構成

使用する全ての端末はSquidに接続し，そこに教師信号付き先読みエージェントを稼働させた．プロキシサーバの構成は図1と同じで，以下の様に動作させた．

実験結果は少ないリクエストの場合に高いヒット率が記録されている．平均ヒット率は38.1%で，単純な先読みによるヒット率の約5.5倍になっている (図 4)．この結果から特定集団ではなく，個人の教師信号をもとにしたキャッシュの先読みにも効果が期待できる．

1. Observer Agletはaccess.logを監視し，教師端末の要求がキャッシュにミスヒットするとAgletsサーバ上でPrefetch Agletを生成する．
2. Prefetch Agletは指定されたリモートサーバへ移動する．
3. Prefetch Agletsは与えられたURLのHTML文から，先読みするリンクを解析する．
4. 先読みするリンクをSquidへ要求する．

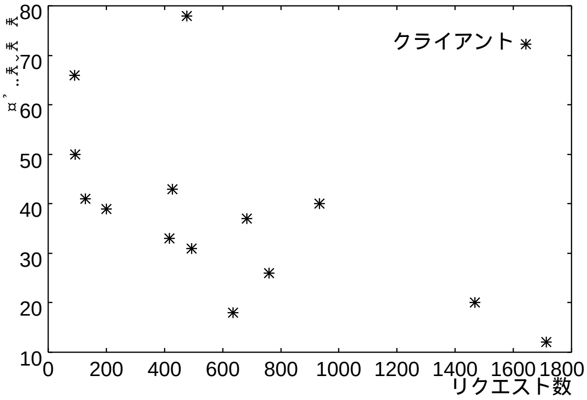

図4 教師信号付き先読みエージェントの実験結果

4 スケーラブルなWWWサービス

WWWでは無数のオブジェクトデータが存在し，それを全てキャッシュすることは不可能である．また単純なキャッシュと先読みは効果が出ないため，クライアントの利用パターンに応じてキャッシュを行う必要がある．そこで本稿では移動エージェントの振る舞いに先読みとクライアントのブックマーク，要求するURLを教師信号としたスケーラブルなWWWサービスを提案する．

実装にはWebサーバ上でHTMLリンク解析を行うWebサーバエージェントと，先読みを行うPrefetchエージェント，クライアントの教師信号を取得するClientエージェントを使用する．

4.1 処理手順

各エージェントは以下の手順で処理を行う(図 5)．実装は図1と同様にSquidに機能を付加する形で行った．Webサーバエージェントにはサーバ上にあるHTMLのリンクを解析する機能，Prefetchエージェントには Squidサーバの親サーバ探索機能，Clientエージェントにはクライアントのブックマークと要求するURLを取得する機能を実装した．

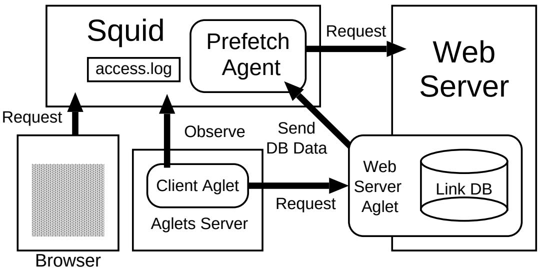

図**5**　システム構成

1. Clientエージェントはクライアントの要求をWebサーバエージェントに問い合わせる．応答が無ければPrefetchエージェントへ直接先読み処理を依頼する．
2. WebサーバエージェントはWebサーバ上のHTML文書のリンクの先読みと重複したリンクの検索を行い，リンク情報データベースの作成と更新を行う．Clientエージェントから問い合わせがあれば，Clientエージェントの指定するProxyサーバエージェントに先読みするリンク情報を通知する．
3. Prefetchエージェントは取得したリンク情報をもとに先読み処理を行う．

4.2　考察

実装したシステムは従来の先読みと比較して，以下の利点が挙げられる．

1. 従来の先読みに比べ，Webサーバエージェントがリンクの先読みと重複するリンクの検索を行うので，先読み処理の分散とリクエストの重複が削減できる．
2. Webサーバ側で資源の参照される頻度を監視することで，Prefetchエージェントにヒットする確率の高いリンク情報を渡すことが可能．
3. WebサーバエージェントがPrefetchエージェントの情報を保持することで，Webサーバ上のHTML文書に変更があった場合にはWebエージェントがPrefetchエージェントに変更データを先読みさせることでWebサーバ側からキャッシュのブロードキャストが可能．
4. Clientエージェントがブラウザのブックマークを監視することで，利用パターンに応じてキャッシュの分散が行える．

5　これからの課題

今回はスケーラブルなWWWサービスとして，先読みと教師信号を採用した移動エージェント間通信と，その処理手順を提案し実装を行った．しかし各エージェントには基本的な機能のみで，多くのユーザが利用する場合には機能の拡張が必要である．

5.1　利用パターンの有効活用

利用するユーザの数が増加すると様々な利用パターンの出現が予測される．実装したPrefetchエージェントはSquidサーバの容量を見てキャッシュの保存先を決定している．これに加えクライアントの利用パターンやキャッシュデータの大きさをもとに先読みするキャッシュの保存先が決定できれば，より効率の良いキャッシュの分散が期待できる．

またClientエージェントの教師信号として，ユーザに表示されているHTMLに含まれるリンクを先読みする手法も考えられる．

5.2　エージェントのインテリジェンス

エージェント間の情報交換は効率良く行わなければならない．一つの手法として参照されるURLに参照回数だけではなく，参照された時間も考慮して重み値を設定する方法もある．この方法だとWebサーバエージェントはPrefetchエージェントへ送るリンク情報に優先順位をつけて効率の良い先読みが可能になる．

同様にClientエージェントはブックマークの利用を監視し，同様な重み値を付ければ利用パターンに応じて先読みする階層の深さを決定できる．

参考文献

[1] “IBM Aglets Homepage”, http://www.trl.ibm.co.jp/aglets/．
[2] “Aglets Software Development Kit”，日本アイ・ビー・エム東京基礎研究所，1998．
[3] 知念 賢一，“Internetにおける大規模情報提供と情報取得の高速化に関する研究”，1998．
[4] “琉球大学総合情報処理センター”, http://www.cc.u-ryukyu.ac.jp/．
[5] “Open System Network”, http://www.osn.u-ryukyu.ac.jp/．
[6] “Squid Internet Object Cache”, http://www.meisei-u.ac.jp/mirror/squid/．