

モバイルエージェントに基づくコンテンツ配信のための キャッシュを考慮した負荷分散手法について

荒井 裕介[†] 近藤 圭祐[†] 大園 忠親[‡] 新谷 虎松[‡]

名古屋工業大学知能情報システム学科[†] 名古屋工業大学大学院工学研究科情報工学専攻[‡]

e-mail: {arai, kondo, ozono, tora}@ics.nitech.ac.jp

1 はじめに

近年、携帯電話は電話やメールなどの簡単な機能しか扱えなかったものが、Web ブラウジングできるようになるまでに急速に発展してきている。しかし、携帯電話のフルブラウザは表示されるまでの速度が遅くユーザにとって不快であるといえる。本システムは、計算機用の Web ページをレイアウトを崩さずに携帯電話上で閲覧できるコンテンツ配信システムである。携帯電話からの要求を受け取ったシステムは、要求された URL 先の Web ページをサーバでレンダリングし、得られた画像と抽出されたリンク情報からモバイルエージェントを作成する。また、作成された Web コンテンツはキャッシュとして保存され、同じ Web ページが要求された場合に再利用する。サーバにとって Web ページのレンダリング処理は高負荷であり、利用するユーザ数の増加に伴って処理速度が遅くなる。

本論文では、モバイルエージェントの作成を複数のサーバへ動的に分担することによるサーバ負荷分散について述べる。

一般的な負荷分散方式であるラウンドロビン方式は、毎回の処理が異なるサーバに割り当てられるため、キャッシュの使用効率が向上しないという問題がある。

2 コンテンツ配信システムの概要

携帯電話は一般の計算機に比べて処理速度、通信速度も低く画像のロードの数が増えれば表示が遅くなる。本システムではデータのロードを減らすため、コンテンツ配信の手法として Web ページをモバイルエージェントのスキームへと変換する。

携帯電話からのリクエストを受け取ったサーバは、リクエストされた URL 先の Web ページをレンダリングし、画像として保存する。さらに、Web ページの情報として、Web ページ中に含まれるリンクの情報を抽出する。Web ページがリンクしている URL のみだけでなく、リンクの表示される場所およびリンクの表示領域を Web ページの情報として抽出して利用する。保存した Web ページの画像と、リンクに関する情報からモバイルエージェントを作成する。この際、作成されたモバイルエージェントはキャッシュとして保存され、再度同じ URL がリクエストされた場合に新規作成するのではなく、キャッシュを再利用することによって処理を省くことができる。

本システムを分散システムとして実現するには、Web ページの画像を分散させるだけでは不十分で、その Web ページの情報として Web ページ中に含まれるリンクや、フォームの情報も分散させる必要がある。したがって、本システムでは、配信するモバイルエージェントを単なる画像でなく、Web ページに関連したプログラムを持ち、クライアントに移動するモバイルエージェントとする。

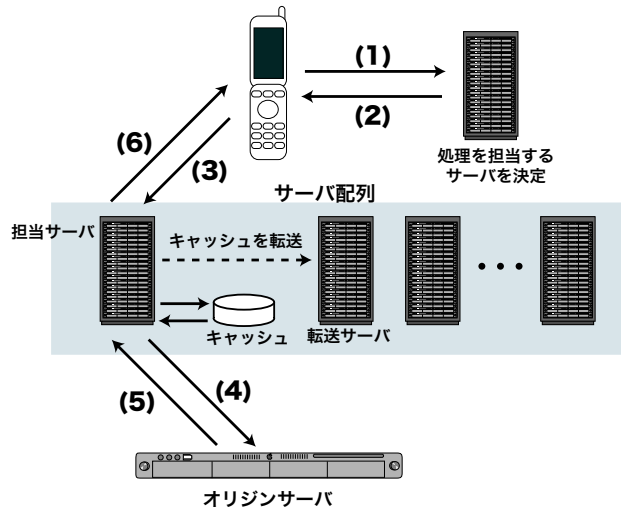


図 1: サーバの構成

3 サーバ負荷分散アルゴリズム

本システムにおいて、サーバに最も負荷をかけるのはモバイルエージェントを作成する処理である。本研究ではサーバを複数用いてモバイルエージェントを作成するサーバを分散させることにより負荷分散を行う。

3.1 サーバの構成

本手法では、携帯電話からの要求を受け取るサーバと、サーバ配列を用いる。サーバ配列とは、Web ページのレンダリングと、モバイルエージェントのキャッシュを保持するサーバの配列である。サーバの構成を図 1 に示す。(1) 携帯電話からの要求を受け取ったサーバはモバイルエージェントを作成せずに、サーバ配列からサーバ（以後、担当サーバと呼ぶ）を 1 つ決める。(2) 携帯電話には (1) で決められた担当サーバのアドレスに転送するページを返す。(3) 携帯電話は、転送ページにより自動で担当サーバにアクセス先を切換えられる。(4) 担当サーバは要求された URL をオリジンサーバに要求する。(5) オリジンサーバから返答されたソースから Web ページをレンダリングし、作成したモバイルエージェントをキャッシュする。(6) 作成したモバイルエージェントを携帯電話に返す。

3.2 処理を担当するサーバの決定

キャッシュの再利用のために、与えられた URL からその URL が示すページを処理するサーバ（担当サーバ）を一意的に決めたい。本手法では、ハッシュ関数を用いてリクエストをユニークなサーバへマッピングする HR(Hash Routing)[?]により、要求された URL とサーバを対応付ける。HR は、単一サーバでの構成と同等のヒット率を得られると共に、サーバ配列内のキャッシュの無駄な重複を避ける。

[†]A Method for Cache-Conscious Load Balancing in a Contents Delivery Based on a Mobile Agent

Yusuke ARAI, Keisuke KONDO, Tadachika OZONO, and Toramatsu SHINTANI

Dept. of Intelligence and Computer Science, Nagoya Institute of Technology, Gokiso, Showa-ku, Nagoya, 466-8555 JAPAN

```

1: 入力: 要求されたURL  $URL$ 
2:   サーバ配列  $Array = \{s_1, s_2, \dots, s_n\}$ 
3: 出力: 処理を担当するサーバ  $S$ 
4: procedure  $DecideServer(URL, Array)$ 
5: begin
6:    $MAX \leftarrow$  ハッシュ関数の値域の最大値;
7:    $n \leftarrow$   $Array$ の長さ;
8:    $hash \leftarrow$   $URL$ のハッシュ値を計算 (MD5);
9:   for  $i \leftarrow 1$  to  $n$  do
10:    if  $hash \leq (MAX * i/n)$  then
11:       $S \leftarrow s_i$ ;
12:      break
13:    endif
14:   return  $S$ ;
15: end.

```

図 2: サーバ決定アルゴリズム

サーバ決定アルゴリズムを図 2 に示す。DecideServer は要求された Web ページの URL (URL) とサーバ配列 ($Array = \{s_1, s_2, \dots, s_n\}$) を入力値とし、モバイルエージェントの作成処理をするサーバ (S) を出力する。まず、 URL からハッシュ関数の 1 つである MD5[3] を用いてハッシュ値 ($hash$) を計算する。次に、MD5 の値域 ($0 \sim 2^{128} - 1$) を $Array$ の長さ (n) で等分した値域をそれぞれのサーバに対応づける。そして、 $hash$ が含まれる値域に対応づけられたサーバ (s_i) を S とする。

ハッシュ関数を用いて計算されるハッシュ値は値域上の偏りがないと言われているため、サーバ配列内のサーバに偏りなく処理を分散できる。

3.3 耐故障性のためのキャッシュコピー

サーバの故障でサーバ配列が変化してしまうと、それまでと異なるサーバで処理をする可能性があり、最大限にキャッシュの再利用ができない。N 台のサーバ配列のうちの 1 台のサーバに故障が発生した場合、URL のハッシュ値をサーバ配列の長さで割った余りと配列番号が等しいサーバに処理をさせるシンプルなハッシュ方式では、故障発生前後で $\frac{N-1}{N}$ も対応付けが変化する。本手法では、故障発生前後の対応付けの変化を $\frac{1}{2}$ 以下に抑えることが可能だが、それでも故障発生時にシステム全体の負荷が増加してしまう。

本手法では、モバイルエージェントを他のサーバに転送し、最小限のキャッシュの重複を許容した DHR (Duplicated Hash Routing)[4] により、故障発生時にもキャッシュを再利用可能とする。

Zipf の法則 [1] によると、ほとんどの Web ページは一度しか要求されないと言われている。ヒットしたときのみキャッシュを転送することで無駄な転送を避けられる。

DecideServer を用いて担当サーバを求めた後、その担当サーバを除いたサーバ配列から再度 DecideServer を用いてサーバ (以後、転送サーバと呼ぶ) を求め、キャッシュの転送先とする。担当サーバに転送サーバのアドレスを伝え、キャッシュがヒットした場合は再利用すると共に、転送サーバへキャッシュをコピーする。また、キャッシュヒットしたものをすべて転送することは無駄であるので、キャッシュを更新した際に作成されたモバイルエージェントを転送する。キャッシュ更新の判断は、要求された URL 先の Web ページのソースの内容に変化がある場合とした。

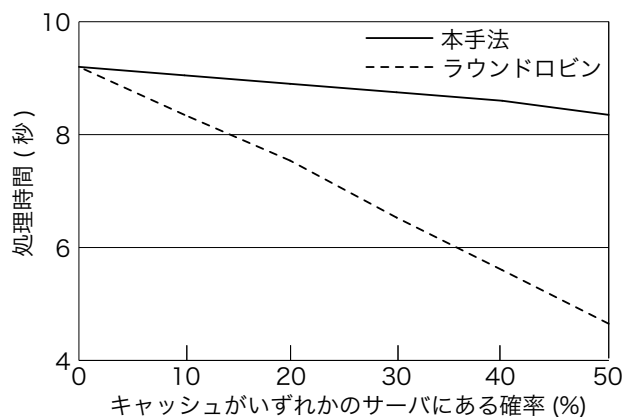


図 3: 本手法とラウンドロビン方式の比較

4 評価

本研究で提案する負荷分散手法では、ハッシュ関数を用いて同じ URL は同じサーバが処理を担当するので、2 度目以降のアクセス時には必ずキャッシュを再利用することができる。本手法の有効性を検証するために一般的な負荷分散方式であるラウンドロビンと比較をする。本論文では、Web ページをモバイルエージェントへ変換する処理をするサーバは 5 台で実験を行った。内容が同じで URL の異なる実験用のページを 500 件用意した。(a) 用意した 500 件の URL すべてを同時接続数の上限を 50 件でシステムに要求し、1 件ごとの平均処理時間を計測した。(b) 次に、500 件のうちの $P\%$ を要求して、サーバ配列内の各サーバにモバイルエージェントをキャッシュさせておいてから、(a) と同様に計測した。(c) キャッシュがいずれかのサーバにある確率 P を 10~50 まで変化させて (b) を繰り返した。(d) 本手法と、ラウンドロビン方式の場合で (a)~(c) の実験を行った。計測結果を図 3 に示す。

5 おわりに

本論文では、計算機用の Web ページをレイアウトを崩さずに携帯電話上で閲覧できるコンテンツ配信システムについて述べ、キャッシュを考慮したサーバ負荷分散の手法を提案した。本手法では、ハッシュ関数を用いて URL とサーバを対応づけることにより、負荷の分散とキャッシュのヒット率を両立させる。また、サーバの故障発生時にもキャッシュのヒット率を維持するためのキャッシュコピーについて述べた。本システムにより、サーバの故障発生時にもキャッシュを有効に再利用できるためより高い耐故障性を実現でき、キャッシュの効率と負荷分散を両立させる。

参考文献

- [1] L. Breslau, P. Cao, L. Fan, G. Phillips and S. Shenker, "Web Caching and Zipf-like Distributions: Evidence and Implications", *IEEE Infocom*, New York, March 1999.
- [2] X. Tang and S. T. Chanson, "Adaptive hash routing for a cluster of client-side web proxies", *Journal of Parallel and Distributed Computing*, vol. 64, No. 10, pp. 1168-1184, October 2004.
- [3] R. Rivest, "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.
- [4] E. Kawai, K. Osuga, K. Chinien and S. Yamaguchi, "Duplicated hash routing: A robust algorithm for a distributed WWW Cache System", *IEICE Transactions on Information and Systems*, vol. E83-D, No. 5, pp. 1039-1047, May 2000.