2003－DBS－131　(36)
2003／7／17

携帯端末のためのキャッシュ方式

鈴木　優[†]　波多野　賢治[†]
吉川　正俊[††]　植村　俊亮[†]

現在，無線ネットワークに接続可能な携帯端末として携帯電話や PDA などの機器が普及し，利用者が必要な情報を閲覧するために用いられている．ところが，携帯端末はネットワーク通信速度が遅いことが多く，利用者が必要な情報を得るために必要な時間は長いことが多い．従来，Web ブラウザの実装においてこれらの問題を解決するために Web キャッシュが用いられてきた．ところが携帯端末では，情報を蓄積しておくべき記憶領域が少ないために，従来の Web キャッシュでは十分な性能が得られない場合がある．これは，従来の Web キャッシュが利用者が情報を利用した時間のみに注目しているために，キャッシュ容量が最適に用いられていないためである．そこで本稿では，携帯端末の利用者が必要な情報をあらかじめ携帯端末に保持しておくための方法として，利用者が情報を利用した時間のみではなく情報の大きさを考慮したキャッシュについての提案を行う．また，キャッシュの性能を測定するための手法としてヒット率が多く用いられてきたが，本稿ではヒット率が十分に携帯端末におけるキャッシュの性能を表しているわけではないことを示し，ヒット率に代わるキャッシュの性能の評価尺度として一つの情報へアクセスするための平均アクセス時間を提案する．

A Caching Method for Portable Computers

Yu Suzuki,[†] Kenji Hatano,[†] Masatoshi Yoshikawa[††] and Shunsuke Uemura[†]

Currently, we have many occasions to use portable computers such as Personal Data Assistants (PDAs) and cellular phones. The wireless communication techniques have evolved rapidly, so that we can access the necessary information from the Internet using the portable computers anytime and anywhere. However, browsing the information generaly takes long time because we usually have to obtain the information from the Internet using the portable computers having a low data transmission speed. In this paper, we propose an information caching algorithm for portable computers. Our algorithm preferentially caches the files, the size of which is large, and is different from current Web caching techniques.

1. はじめに

計算機の小型化が進むにつれて，人は計算機をどこへ行くにも持ち歩き，利用することができるようになっている．現在の実用的な計算機の大きさはラップトップコンピュータだが，今後はさらに小型化が進み，近い将来，ウェアラブルコンピュータのようにいつでもどこでも計算機を利用できる環境が整うだろうと考えられる．

利用者が携帯端末を用いて情報をネットワーク上から取得する際の最も大きなボトルネックは，ネットワークの性能である．利用者が遅いネットワーク上にある情報を要求した時には，利用者はその情報が利用者の端末まで送られてくるまで待たなければならない．ところが，携帯端末の持つネットワークデバイスは一般に非常にネットワーク速度が遅いため，利用者は非常に長い時間待つ必要がある．そこで，利用者が情報にアクセスするための速度を向上させるために，利用者が頻繁に利用する情報をあらかじめ端末に保持しておくことが考えられる．インターネット上での World Wide Web (WWW) では，アクセス速度の改善のために Web キャッシュを用いている．そのため，Web キャッシュは重要な研究テーマとなりつつある．

利用者の利用する情報を全てキャッシュすることは，利用者の端末の持つ端末が持っているキャッシュ容量が有限であるために不可能であるといえる．そのため，従来のキャッシュに関する研究では，どのようにして有限のキャッシュ容

† 奈良先端科学技術大学院大学 情報科学研究科
Graduate School of Infomation Science, Nara Institute of Science and Technology
†† 名古屋大学 情報連携基盤センター
Information Technology Center, Nagoya University

量を有効に用いることができるかが課題となっている．現在提案されている方法は LRU (Least Recent Used) がベースとなっている．

LRU とは，キャッシュに入っている情報のうち最も長い間アクセスされていない情報から順にキャッシュから削除する方法である．ところが，LRU では利用者が情報にアクセスした時間のみを基準に記憶容量にデータを残すかどうかを決めていた．だが，利用者の利用する情報は全て同じ大きさではない．よって，情報のサイズの大小に関わらずキャッシュに入れてしまうために，もし小さな情報ばかキャッシュに入ってしまった場合に利用者が大きな情報を得るために長い時間待ならないという問題がある．そこで，情報の大きさも考慮したキャッシュの方法を利用することによって，よりキャッシュ容量を最適に使うことができると考えられる．

そこで本稿では，情報のアクセス時間に加えて情報のサイズも考慮した携帯端末のためのキャッシュの構築手法についての提案を行う．本稿で提案する手法は，情報のサイズの大きなものを優先的にキャッシュに残す方法である．なぜなら，従来の Web キャッシュの方法では利用者のネットワークへの総アクセス時間を減らすことを目標としていたが，我々の手法では利用者は情報一つあたりのアクセスする時間をより重要視しているからである．つまり，情報一つあたりの転送時間を下げるためには大きな情報を優先的にキャッシュしておいたほうがよりキャッシュを有効に使用していると考える．これは，Web キャッシュにおける考え方である，小さな情報を優先的にキャッシュしておいたほうがヒット率が上がりキャッシュを有効に使用しているという考え方とは全く逆の考え方である．そこで，Web キャッシュの場合であれば，ヒット率を上げるためにサイズの小さい情報を優先してキャッシュに入れる方針であることに対して，我々の提案する携帯端末のためのキャッシュ方法では，サイズの大きい情報を優先してキャッシュに入れる．

また，本稿ではヒット率が携帯端末におけるキャッシュの性能を測定するための方法として適切ではないことを示す．従来，キャッシュの性能を測定するためにヒット率が用いられてきたが，ヒット率を上昇させることが必ずしもキャッシュの性能が上がったことを示すわけではない．また，データアクセスの総時間を減らす方法もキャッシュの性能をあらわすために用いているが，携帯端末のためのキャッシュ性能を測定する方法としては不適である．なぜなら，我々の提案するキャッシュの目標が総アクセス時間の短縮ではなく，一つあたりのアクセス時間の短縮にあるからである．

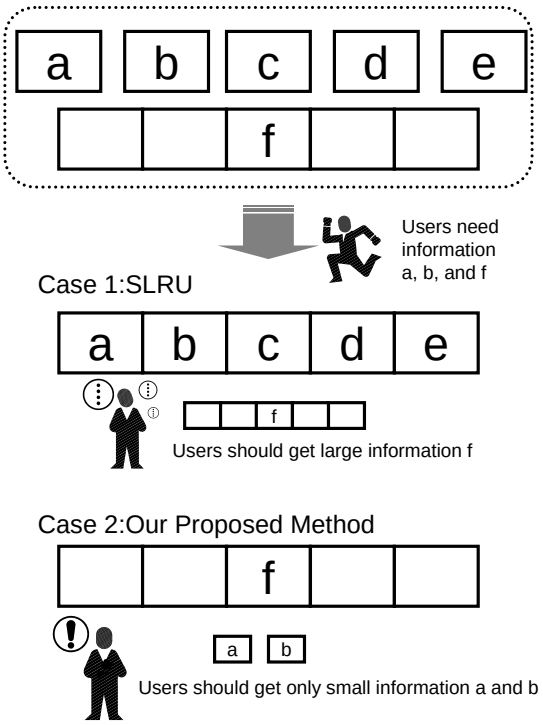

図 1　Size Adjusted-LRU と提案手法との比較
Fig. 1　Size Adjusted-LRU vs. our proposed method.

2. 関 連 研 究

WWW では，多くの場面でキャッシュ技術が用いられている．最も良く用いられているキャッシュとして代理サーバ (Proxy Server)[1] が挙げられるが，その他にも Web ブラウザや Web サーバなどでもキャッシュ技術は用いられている[2),3)]．これらのキャッシュの主な目的は，WWW 上におけるネットワーク上のデータの流通量を減少させ，かつ利用者が情報を得るための待ち時間を減少させることである．

1 章でも述べたように，Web キャッシュの方法における基本的な戦略は，有限のキャッシュ容量をいかに有効利用するのかにある．従来の Web

キャッシュ技術で用いられる方法である LRU は最も単純な戦略であるが，情報のサイズが考慮されていない点が問題である．そのため，情報のサイズの大小に関わらずキャッシュに入れる．ところが，サイズが小さい情報はキャッシュされていない場合でもネットワークを使用して短時間で取得できる．つまり，サイズの大きな情報をキャッシュに入れた場合，サイズの小さな情報をキャッシュに入れた場合に比べてより利用者が情報を得るための待ち時間を減らすことができると考えられる．

一方，情報のサイズに注目したキャッシュ技術も提案されている[4]．この方法では，キャッシュされた情報のうち，サイズ大きなものから順にキャッシュから削除する．この方法ではキャッシュの中に大きなサイズの情報のみが残ることはない．ところが，アクセス時間を基にした情報は用いていないため，長い間アクセスされておらず，小さいサイズの情報がキャッシュに残ってしまう．そのため，利用者が以前閲覧した大きなサイズの情報を再び閲覧する場合には，利用者は長い時間待たなければならない．

こうした問題点を解決するために，LRU とキャッシュされる情報のサイズの両方に注目したキャッシュ技術が，Aggrawal らによって提案された Size Adjusted-LRU (SLRU)[5] である．SLRU では，Web ページをキャッシュする際に，サイズの小さな情報でかつ最後に閲覧した時間からの経過時間が短い情報をキャッシュすべき情報であるという戦略を採用している．具体的には，最後にある情報 O にアクセスした時間からの経過時間を $\tau(O)$，情報のサイズを $\sigma(O)$ とすると，情報 O のスコアは $\tau(O)\cdot\sigma(O)$ で計算され，このスコアの大きいものから順にキャッシュから削除してゆく．しかし，SLRU は，Web キャッシュと同様，ネットワーク通信速度とキャッシュ容量を十分に持った計算機に対して有効な方法であり，携帯端末に同じ方法を用いた場合には有効な方法であるとは限らない．なぜなら，SLRU はヒット率を向上させるためにサイズの小さな情報を優先的にキャッシュへ格納するような戦略を採用しており，例えば，ほぼ同じ時間にアクセスした 2 つの情報があり，1 つのサイズは小さくもう 1 つは大きいような場合，大きなサイズの情報はキャッシュサーバに格納されず，利用者が再度その情報を所望した場合，その情報を閲覧するまでに長い待ち時間を必要とするからである．

SLRU と我々の提案手法との違いを，図 1 を用いて述べる．サイズが 1 である情報 a, b, c, d, e がキャッシュされている場合とサイズが 5 である情報 f だけがキャッシュされている場合．利用者が再利用した情報が a, b, f であった場合，ヒット率の観点から考えると前者のアルゴリズムのほうが優れているが，閲覧するまでの必要時間を考えた場合は後者のアルゴリズムのほうが優れている☆．言い換えれば，Web キャッシュではサイズの小さい情報をキャッシュに格納する戦略をとるのに対し，携帯端末におけるキャッシュ方法ではサイズの大きな情報をキャッシュに格納する戦略を採用する．

3. 携帯端末のためのキャッシュの方法

以下，オブジェクトとは利用者が閲覧する一つの情報を指し，例えば 1 つの Web 文書や PDF 文書がオブジェクトに相当する．

利用者があるオブジェクトを閲覧する時，次の手順を用いてキャッシュに含まれるオブジェクトの操作を行う．

(1) 利用者があるオブジェクト O を所望する．
(2) システムはキャッシュにオブジェクト O が格納されているかどうかを確認する．
- オブジェクト O がキャッシュに存在する場合には，キャッシュから O を抽出して利用者に提示する．
- オブジェクト O がキャッシュに存在しない場合には，ネットワーク上からオブジェクト O を取得後利用者に提示し，キャッシュに格納する．格納の際，キャッシュ領域に空き容量がない場合，キャッシュ領域から最も不要なオブジェクトを見つけ消去する．

以上の流れからも分かるとおり，キャッシュを行うための方法として最も重要なことは，キャッシュ領域に格納されているオブジェクトのうち，最も不要なオブジェクトをどのように発見するかという点である．我々の提案する方法では，次のような手順で最も不要なオブジェクトを消去

☆ サイズ 1 の情報を取得するのに時間 1 かかるとした場合，携帯端末で閲覧できるまでの必要時間は，前者は時間 5，後者は時間 2 である．

する．

(1) オブジェクト O の最終アクセス時間から計算されるスコア $\tau(O)$ を計算する．
(2) オブジェクト O のサイズから計算されるスコア $\sigma(O)$ を計算する．
(3) $\tau(O)$ と $\sigma(O)$ を統合し，オブジェクト O のスコア $r(O)$ を計算する．
(4) キャッシュに含まれる全てのオブジェクトに対してスコア $r(O)$ を計算し，最もスコア $r(O)$ の値が低いものから順番にキャッシュから消去する．

以下では，オブジェクトに与える 2 つのスコアの定義方法について述べ，次にそれら 2 つのスコアを統合するための方法について述べる．

3.1 スコアの計算方法

オブジェクト O へのアクセス時間のスコア $\tau(O)$ やファイルサイズのスコア $\sigma(O)$，それら 2 つのスコアを統合したスコア $r(O)$ は，全て値が小さければ小さいほどキャッシュに残すべきであることを示している．

3.1.1 最終アクセス時間から求めるスコア

オブジェクト O へのアクセス時間から計算されるスコアを $\tau(O)$ とおき，次のように定義する．$\Delta(T(O))$

$$\tau(O) = \frac{d\_rank(\Delta T(O))}{N} \tag{1}$$

ここで，$\Delta T(O)$ はオブジェクト O が最後にアクセスした時間からの経過時間である．また，関数 d_rank は，キャッシュ内に含まれるすべてのオブジェクトに対して $\Delta T(O)$ を計算し，降順で並び替えた場合の順位である．つまり，LRU と同様，利用者が一番最近アクセスされたオブジェクトが最もスコアが高くなる．また，最も長い間アクセスされなかったオブジェクトはスコアが低くなり，そのため最もキャッシュから捨てられやすくなる．

3.1.2 オブジェクトのサイズから求めるスコア

オブジェクト i のファイルサイズ $S(O)$ から計算されるスコアを $\sigma(O)$ とおき，次のように定義する．

$$\sigma(i) = \frac{a\_rank(S(O))}{N} \tag{2}$$

関数 a_rank は，キャッシュ内に含まれる全てのオブジェクトのスコア $\sigma(O)$ を昇順で並べた時の順位である．

ここで，オブジェクトのサイズ $S(O)$ が小さいものほどスコアが低くなる点が，提案手法における重要な点の一つである．従来の Web キャッシュではオブジェクトのサイズが小さいほどスコアが大きくなるように関数を設定するため，サイズの小さいものほどキャッシュに残る．本研究ではサイズの大きなオブジェクトをキャッシュに残すため，オブジェクトのサイズが小さいものほどスコアを低くする．

3.2 統 合 手 法

最後に，$\tau(O)$ や $\sigma(O)$ を用いてオブジェクト O のスコア $r(O)$ を求める．

$$r(O) = \begin{cases} 1 - \left(\frac{(1-\tau(O))^p + (1-\sigma(O))^p}{2}\right)^{\frac{1}{p}} \\ \left(\frac{\tau(O)^p + \sigma(O)^p}{2}\right)^{\frac{1}{p}} \end{cases} \tag{3}$$

2 つの値を統合する際には 2 つの式のどちらか一方を用いる．キャッシュに含まれる全てのオブジェクトに対して $r(O)$ を求め，最も $r(O)$ が小さなものから順にキャッシュから消去する．p $(1 \leq p)$ は 3 式のパラメータである．これら 2 つの式は p-norm と呼ばれており，情報検索において複数の値を統合するための式として用いられている．3 式の上段は p-norm のうち “and” を示すものであり，下段は “or” を示す．and とは，2 つの値のどちらもが高い値である場合に統合スコアが高い値となり，or とは 2 つの値のどちらかが高い値である場合に統合スコアが高い値となる．

SLRU では $\tau(O)$ と $\sigma(O)$ を統合する際に関数として積を用いている．ところが，我々の以前の研究[6] では，複数のスコアを統合して一つの値を求める際の関数として和や積など一般的な関数では性能が低下する場合があることが実際に示されている．そのため，提案手法の場合のように 2 つのスコアを統合する場合には関数の選択は重要であると考えられる．複数の値を統合するための関数として平均や最大値，最小値などが用いられることが多いが，これらは p-norm によってパラメータ表示ができる．そこで，$\tau(i)$，$\sigma(i)$ を統合するための関数として p-norm を用いた．

3.3 空きキャッシュ容量の有効活用

提案手法では，3.2 節の方法を用いてスコアの最も低いオブジェクトから順にキャッシュから消去するが，この消去方法では空いた容量が無駄となってしまう恐れがある．例えば，キャッシュ容量として 7 の容量があり，キャッシュにはそれぞれ大きさ 1，2，4 の 3 つのオブジェクト o_1, o_2, o_4 があったとする（図 2）．ここで，利用者

が大きさ 5 のオブジェクト o_5 を要求したと仮定する．o_5 をキャッシュの中に入れた場合，キャッシュに含まれるオブジェクトのサイズの合計は 12 となってしまい，キャッシュの容量 7 を超えてしまうため，提案したアルゴリズムを用いてキャッシュに含まれるオブジェクトを消去しなければならない．簡単のため，キャッシュに含まれていた 3 つのオブジェクト o_1，o_2，o_4，o_5 のアクセス時間によるスコア $\tau(i)$ は同じであると仮定する．すると，o_5 以外のオブジェクトサイズが小さなものから順に消去される．つまり，まず o_1 が消去されることになる．ところが，o_1 を消去してもキャッシュ容量よりもオブジェクトのサイズのほうが大きいために，次にスコアの大きい o_2，o_4 を順に消去する．ここで，o_5 をキャッシュに入れてもキャッシュの容量は十分あるため，ここでキャッシュの消去の処理は終了する．つまり，最終的にはキャッシュの中にはオブジェクト o_5 が残り，大きさ 2 の空き容量が残ることになる．

ところが，最後に残っている大きさ 2 の空き容量の中には o_1 や o_2 を入れることが可能であり，この場合では o_2 と o_5 が残るようなアルゴリズムが望ましいと考えられる．そこで，次に示した方法を用いてキャッシュへオブジェクトを戻す．

(1) 3.2 節で求められたスコアのうち最も小さなものを求め，そのオブジェクトを消去する
(2) (1) を用いた結果，空いたキャッシュのサイズ S_j を求める．
(3) オブジェクトのサイズが S_j を超えないもののうち，スコアの大きなオブジェクトから順にキャッシュへ戻す．

本節の例の場合は，o_1 から o_5 のうちで最もスコアが高いオブジェクトは o_5 であるため，キャッシュの空き容量に入れることができないオブジェクト o_4 を除くと最もスコアの高いオブジェクトは o_2 である．そこで，o_5 と o_2 のみがキャッシュに残ることとなり，空きキャッシュ容量が有効に用いることができると考えられる．

4. 提案手法の評価方法

我々の提案した手法は，利用者がオブジェクトにアクセスするために必要な時間を削減するために必要なキャッシュの方法である．そこで，実際に提案手法によってどの程度オブジェクトにアクセスするために必要な時間が削減できているかを示すために，評価実験を行う．キャッシュの方法が有効であることを示すために従来からヒット率と呼ばれる手法が利用されていたが，携帯端末のキャッシュに適していないと考えられる．そこで，本節ではまずヒット率が携帯端末のキャッシュを評価するために適していない根拠を示し，提案手法を評価するために適している評価尺度として平均アクセス時間を提案する．最後に，我々の提案が Web キャッシュと比較して，携帯端末のためのキャッシュ法として優れていることを示すための手順について述べる．

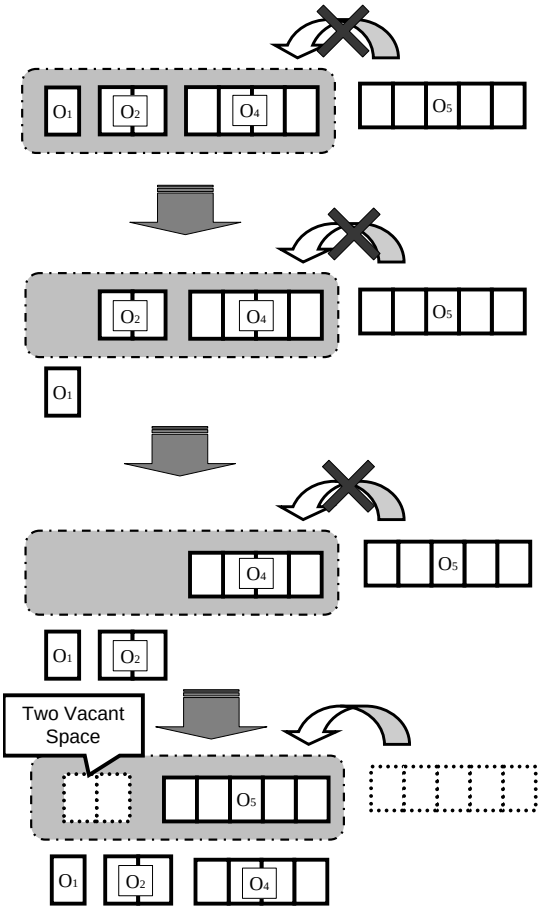

図 2 空きキャッシュ容量を有効に活用するための方法
Fig. 2 A Strategy of the Vacant Cache Fulfillment.

4.1 携帯端末の評価尺度としてのヒット率の限界

我々の提案手法は，目的はヒット率を向上させ

ることではなく，ある情報を閲覧するために必要な時間を短縮することにある．これは，ヒット率が携帯端末におけるキャッシュの性能を評価できているとはいえないからである．そこで，まず，ヒット率を用いることがキャッシュの性能を評価することに限界があるという点について，2 章の図 1 を用いて SLRU[5] と提案手法を比較することによって示す．

まず，6 つのオブジェクト a，b，c，d，e，f を仮定し，オブジェクト a，b，c，d，e の大きさは 1，オブジェクト f の大きさは 5 であるとする．また，サイズ 1 のオブジェクトがキャッシュ上に存在しなかった場合に，ネットワークを用いてオブジェクトの転送を行い利用者が閲覧するために時間 1 かかるとする．逆に，もしキャッシュ上にオブジェクトが存在した場合には，利用者が閲覧するための時間はかからないとする．また，ここではオブジェクトのサイズのスコアがキャッシュの性能に与える影響のみに注目しているため，最終アクセス時間から計算されるスコアは全てのオブジェクトに対して同じであるとする．

以上の仮定を用いて，利用者が SLRU，提案手法それぞれを用いてキャッシュを行う．まず，利用者は 6 つのオブジェクト全てを閲覧する．この時，SLRU を用いてキャッシュを行った場合には，小さいオブジェクトから優先してキャッシュに格納されるため，キャッシュの中には 5 つのオブジェクト a，b，c，d，e が格納される．一方，我々の提案手法を用いてキャッシュを行った場合には，大きなオブジェクトから優先してキャッシュに格納されるため，キャッシュの中には 1 つのオブジェクト f が格納される．

次に，利用者がオブジェクト a，b，f を閲覧するために必要な時間を考える．SLRU を用いてキャッシュを行った場合には，a, b が既にキャッシュに格納されているために，オブジェクト a と b を閲覧するために必要な時間は 0 である．ところが，オブジェクト f はキャッシュに格納されていないため，ネットワークを用いてオブジェクトの転送を行う必要がある．つまり，オブジェクト f の大きさが 5 であるため，時間 5 がかかる．

一方，我々の提案手法を用いてキャッシュを行った場合には，f が既にキャッシュに格納されているために，オブジェクト f を閲覧するのに必要な時間は 0 である．ところが，オブジェクト a と b はキャッシュに格納されていないため，時間 1 づつ，合計時間 2 をかけてオブジェクト a，b を転送しなければならない．

ここで，キャッシュの性能を測るための一般的な指標であるヒット率を用いてそれぞれの方法を比較する．ヒット率 H は，次の式を用いて求めることができる．

$$H = \frac{c_h}{n_h} \tag{4}$$

n_h は利用者が閲覧した全てのオブジェクトの数であり，c_h は利用者が閲覧した全てのオブジェクトのうち，キャッシュに格納されていたオブジェクトの数である．SLRU を用いた場合のヒット率は，利用者が閲覧した全てのオブジェクトが，まず最初に閲覧したオブジェクト a，$\cdots$，f の 6 つのオブジェクトと利用者が再度見たオブジェクト a，b，f の 3 つであり，合計 9 個である．一方，キャッシュに格納されていたオブジェクトの数は a，b の 2 つであるため，ヒット率 H は $\frac{2}{9} = 0.22$ となる．一方，我々の提案した手法はキャッシュに格納されていたオブジェクトの数が f の 1 つであり，ヒット率は $\frac{1}{9} = 0.11$ となる．つまり，我々の提案手法は SLRU と比較して劣っていることになる．

ところが，携帯端末のためのキャッシュの目的とは利用者がオブジェクトを閲覧するために必要なネットワーク転送時間を下げることが目的である．そこで，全てのオブジェクトを閲覧するために必要な時間を評価尺度として SLRU と我々の手法を比較する．SLRU では，全てのオブジェクトを転送するために，a，$\cdots$，f の 6 つのオブジェクトと再度転送しなければならない f をネットワークで転送する時間がかかる．この場合，一度閲覧したオブジェクトを再度閲覧するために 5 の時間がかかる．一方，我々の提案手法では a，$\cdots$，f の 6 つのオブジェクトと a，b の 2 つのオブジェクトをネットワークで転送する時間がかかるため，再度オブジェクトを取得するために 2 の時間がかかる．つまり，SLRU よりも我々の提案手法はネットワーク転送時間は少なく，良いことが分かる．

ヒット率はオブジェクトの大きさを考慮していないため，オブジェクトの大きさがある程度同じ場合に有効な指標であると考えられる．ところが，携帯端末で閲覧することのできるオブ

ジェクトは，非常に小さいものから非常に大きなものまで様々な大きさのものが存在する．そのため，本稿で提案する手法を評価するための手法としてヒット率は有効であるとはいえない．

本稿では，それぞれの大きさが異なるオブジェクトのためのキャッシュ手法の評価尺度として，一つのオブジェクトを取得するための単位あたりの平均取得時間を用いることとした．この評価尺度を用いることにより，オブジェクトの大きさを考慮したキャッシュ手法の評価を行うことができる．

4.2 オブジェクトの大きさを考慮した携帯端末のためのキャッシュの評価尺度

4.1 節で述べた通り，ヒット率は携帯端末のキャッシュの性能を示していないことが分かり，かつオブジェクトの平均アクセス時間を用いて携帯端末のキャッシュの性能を明らかにできることを示した．そこで，携帯端末のキャッシュの性能を評価するための指標として平均アクセス時間による評価尺度について示す．$t(O)$ はオブジェクト O を利用者が要求した時に必要であったアクセス時間であり，キャッシュにオブジェクト O がある場合のアクセス時間 $t(O)$ を次のように定義する．

$$t(O) = \sigma(O) \cdot L \tag{5}$$

L は携帯端末の通信速度であり，簡単のために定数とする．また，キャッシュに O_i が存在した場合のアクセス時間は，携帯端末に存在する記憶容量へアクセスするための時間である．ここで，携帯端末が記憶容量にアクセスする時間はネットワークへアクセスする時間と比較すると無視できるほど小さいため，L はほぼ 0 に近似できる．そこで，以下で定義するように，1 つのオブジェクトを利用者が閲覧するまでの平均アクセス時間を用いて提案手法の評価を行う．

$$E = \frac{\sum t(O)}{n} \tag{6}$$

ここで，E は平均アクセス時間を示し，n は利用者が要求した総オブジェクト数である．

以上の定義を用いて，キャッシュの容量に対応する平均アクセス時間 E を求める．

4.3 実験手順

以下に示す実験手順を用いて，提案手法と SLRU，LRU との平均アクセス時間による比較を示す．

(1) 一人の利用者が WWW ブラウザで閲覧した HTML 文書の閲覧履歴を取得する．例えば，利用者が利用している Proxy サーバなどの利用履歴を取得することなどが考えられる．また，携帯端末は基本的に 1 人で使用すると考えられるため，1 人の閲覧履歴を実験対象とする．

(2) 実験として想定するキャッシュの容量を求める．キャッシュの効果は，利用者が利用した全オブジェクトの容量におけるキャッシュ容量の割合に比例すると考えられる．そのため，まず利用者が利用した全オブジェクトの容量 C を求める．そして，C の 0.5% から 20% をキャッシュの容量 C_s として設定する．

(3) キャッシュ容量 C_s と平均アクセス時間 E との相関関係についての調査を行う．提案手法と SLRU，LRU を比較して同じ条件下での実験を行った上で，特にキャッシュの容量が小さい時の平均アクセス時間に注目する．なぜなら，携帯端末には記憶容量が小さく，キャッシュの容量も小さいことが考えられるため，キャッシュの容量が小さい場合に有効なキャッシュの方法が携帯端末において有効なキャッシュ方式であると考えられるからである．

5. おわりに

本稿では，携帯端末を用いて情報を閲覧するために必要なキャッシュの方法とその評価法の提案を行った．我々の提案するキャッシュ方法は次のような特徴をもつ．

- 携帯端末のためのキャッシュ技術では，キャッシュに利用できるディスク容量とネットワークの速度を考慮する必要がある．本研究ではネットワークの速度の遅さを改善するために，サイズの大きい情報を優先的にキャッシュに格納するアルゴリズムを提案した．
- キャッシュに空きがなくなった場合に，キャッシュに含まれる情報から最も不要な情報を消去する必要があるが，その際に 2 つの尺度，情報のサイズと LRU から算出されるスコアを統合利用した．

以下の点は，今後の研究課題である．

- 本論文で提案したアルゴリズムを携帯端末

におけるキャッシュとして実装し，実際に提案手法が有効であることを示さなければならない．

- 本研究ではパラメータを含む関数を用いて 2 つのスコアを統合したが，パラメータによってキャッシュの性能が上下することが考えられる．つまり，最適なパラメータを用いることによってキャッシュの性能を上げることができると考えられるため，自動的にパラメータを決める方法を考えなければならない．
- 2 つのスコアを統合する際に，本論文では 2 つのスコアを平等に評価をしている．ところが，オブジェクトのサイズ，閲覧時間のどちらか一方をより重視した場合によりキャッシュの性能が上昇すると考えられる．そこで，2 つのスコアそれぞれに重みを与えた上で統合する方法が考えられるが，重みを利用者が直接指定することは非常に難しい．そこで，重みを自動的にシステムが設定する方法について考えなければならない．
- 3.3 節における方法は，本研究における目的である 1 つのオブジェクトあたりのオブジェクト取得時間を短縮するという目的のために行っているわけではなく，従来のキャッシュの目標であるヒット率の向上のために行っている．本研究で提案した手法は，従来のキャッシュ手法と比べてヒット率が低下することが考えられるが，ヒット率が低下することによってキャッシュの性能として低下すると考えられる．そこで，ヒット率を低下させることなくオブジェクト取得時間を短縮する方法について考えなければならない．

謝辞　本研究の一部は，文部科学省科学技術研究費補助金（課題番号：14780325），ならびに科学技術振興事業団 (JST) の戦略的基礎研究推進事業 (CREST)「高度メディア社会の生活情報技術」プログラムの支援によるものである．ここに記して感謝を表す．

参 考 文 献

1) Luotonen, A. and Altis, L.: World-Wide Web Proxies, *Proc. First Int'l World Wide Web Conf.* (1994).
2) Abrams, M., Standridge, C., Abdulla, G., Williams, S. and Fox, E.: Caching Proxies: Limitations and Potentials, *Proc. Fourth Int'l World Wide Web Conf.* (1995).
3) Braun, H. and Cla, K.: Web Traffic Characterization: An Assessment of the Impact of Caching Documents from NCSA's Web Server, *Proc. Second Int'l World Wide Web Conf.* (1994).
4) Williams, S., Abrams, M., Standridge, C., Abdulla, G. and Fox, E.: Removal Polices in Network Caches for World Wide Web Documents, *Proc. ACM SIGCOMM*, pp. 293 – 304 (1996).
5) Aggrawal, C., Wolf, J.L. and Yu, P.S.: Caching on the World Wide Web, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, No. 1, pp. 94 – 107 (1999).
6) Suzuki, Y., Hatano, K. and Uemura, S.: A Calculation Method of Document Scores for Multimedia Document Retrieval, *Proc. IASTED Int'l Conf. on Information Systems and Databases (ISDB 2002)*, pp. 178 – 183 (2002).