スループット評価指標を導入したWWWキャッシュ置き換えアルゴリズムの提案と評価

長田智和[†] 神里大[†]
谷口祐治[††] 玉城史朗[†]

近年，World Wide Web（以下，WWW）を用いたインターネット上での情報サービスが急速に普及し，ネットワークやサーバの負荷増大によってWWWサービス応答時間の遅延が問題となっている．また，ネットワークやサーバの負荷増大は，複数のユーザが同一ページを閲覧することでさらに助長されている．これらの問題を解決するため，WWWキャッシュサーバが広く利用されている．一方，従来のWWWキャッシュサーバで用いられているページデータの管理（キャッシュへの保存や削除）のためのキャッシュ置き換えアルゴリズムは，アクセス頻度やデータサイズから主にヒット率やバイトヒット率の改善を目的とした手法は多いが，クライアントが要求したページデータを受信するのに要するサービス応答時間が明確に改善された有効な手法はない．そこで，本論文では，アクセス頻度とデータサイズに加えて，スループットを考慮することで，サービス応答時間を改善する新たなキャッシュ置き換えアルゴリズムを提案する．提案手法をWWWキャッシュサーバに実装して評価を行った結果，従来手法と比べてサービス応答時間の改善を確認した．

A WWW Cache Replacement Algorithm Utilizing Throughput Evaluation Indexes

Tomokazu Nagata,[†] Masaru Kamizato,[†] Yuji Taniguchi[††] and Shiro Tamaki[†]

In recent years, information services on the Internet using World Wide Web (WWW) spread rapidly and delay of WWW service response time is caused by increase of network traffic and server load simultaneously. The loads on network and server are encouraged by redundancy of requesting same pages by many people, even when they browse the same pages. In order to solve the redandancy, WWW cache server is used broadly. However, most of existing cache replacement algorithms for administration of page data, used by WWW cache server to choose which data to be evicted from the cache, aim at only improving cache hit rate and byte hit rate based on such data as access frequency and data size of each cache object. There is no effective algorithm that explicitly intends to improving service response time. In this paper, I propose a new cache replace algorithm to improve service response time by considering throughput in addition to access frequency and data size. The result of implementing the method in a WWW cache server proved its effectiveness.

1. はじめに

近年，WWWを用いたインターネット上での情報サービスが急速に普及している．その結果，ネットワークやサーバの負荷が増加し，WWWサービス応答時間の遅延が問題となっている．また，ネットワークやサーバの負荷増大は，複数のユーザが同一ページを閲覧することでさらに助長されている．これらの問題を解決するため，WWWキャッシュサーバが広く利用されている．

WWWキャッシュサーバの基本的な動作概要は次のとおりである．まず，クライアントはWWWキャッシュサーバにページデータの要求を行う．次に，WWWキャッシュサーバは自身のキャッシュに要求されたページデータが保持されていれば，キャッシュから該当するページデータをクライアントに転送し，逆に，キャッシュに保持されていなければ，WWWサーバからページデータを取得してクライアントに転送するとともに

† 琉球大学理工学研究科総合知能工学専攻
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

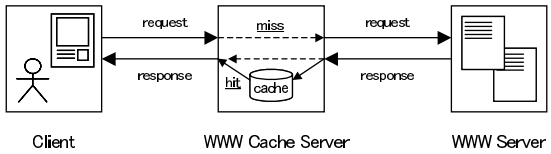

図 1 WWW キャッシュサーバの動作イメージ
Fig. 1 An image of WWW cache server.

キャッシュ領域にも格納する（**図 1**）．以上のようなキャッシュ機能により，キャッシュしたページデータの再利用が可能となり，WWW サーバと WWW キャッシュサーバ間の通信量が削減されるだけでなく，サービス応答時間も短縮される．

一方，WWW キャッシュ技術の研究もさかんに行われている．特に，キャッシュ溢れが起こった場合のページデータの管理（キャッシュへの格納や削除）のためのキャッシュ置き換えアルゴリズムの代表例としては，ページデータのアクセス頻度やデータサイズを考慮することで，ヒット率やバイトヒット率の改善を目的とした手法[1),2)]が提案されている．さらには，ページデータの取得時間を考慮することで，サービス応答時間の改善に着目した手法[3)]や，アクセス頻度やデータサイズ，ページデータの取得時間など複数の指標を考慮することで，複合的な性能改善を目的とした手法[4)~6)]も提案されている．ここで，後者は前者のヒット率やバイトヒット率の改善を目的とした手法に比べて，サービス応答時間を改善する反面，ヒット率やバイトヒット率が振るわないなど，性能改善に偏りがあるものが多い．たとえば，文献 5) の提案手法（GreedyDual 方式）では，データサイズともう 1 つの評価指標の組合せによってキャッシュ優先度を求めることでサービス応答時間を改善しているが，アクセス頻度を考慮していないために明示的にヒット率を改善する手法ではなく，また，バイトヒット率の点からも，他の手法に対して優位性がある手法ではない．

そこで，本論文では特にサービス応答時間の改善☆を目的とし，さらにヒット率やバイトヒット率の改善も目指した WWW キャッシュサーバを実現するため，従来，キャッシュ優先度を決定する評価指標として用いられていたアクセス頻度とデータサイズに加えて，スループットを考慮した新たなキャッシュ置き換えアルゴリズムを提案する．ここで，サービス応答時間をクライアントが WWW キャッシュサーバにページデータを要求してページデータを受信するまでに要する時間とし，スループットを単位サービス応答時間あたりの転送データサイズと定義すると，スループットを評価指標に導入することにより，サービス応答時間の改善が期待できる．また，アクセス頻度，データサイズを考慮することにより，ヒット率およびバイトヒット率の改善も可能であると考えられる．提案手法を WWW キャッシュサーバに実装して評価を行った結果，従来手法と比べてサービス応答時間の改善を確認した．また，従来手法と比べてヒット率は同等であり，各評価指標の重み付けの変更によってバイトヒット率も同等の改善結果が得られた．

本論文では，まず，2 章で WWW キャッシュサーバのログから WWW アクセス分析を行う．次に，3 章ではアクセス頻度，データサイズ，スループットの 3 つの評価指標からページデータのキャッシュ優先度を決定する評価式を導出し，この評価式による新たなキャッシュ置き換えアルゴリズムを提案する．さらに，提案手法を WWW キャッシュサーバに実装して性能評価を行う．最後に，4 章でまとめを述べる．

2. WWW アクセスの特徴分析

まず，サービス応答時間を改善するキャッシュ置き換えアルゴリズムにおいて，考慮すべきページデータの評価指標を見い出すため，我々が運用している WWW キャッシュサーバのアクセスログの分析を行った．分析対象の WWW キャッシュサーバは琉球大学で学内向けに公開運用しているものであり，過去 3 カ月分（2002/4/1〜2002/6/30）約 18 万リクエストのログを分析した．また，純粋な WWW アクセスの特徴分析を目的とするため，サービス応答時間（**図 4**）およびスループット（**図 5**）のデータは 1 回目のリクエストのみを分析対象とし，キャッシュヒットしたリクエストは除外した．リクエスト回数（**図 2**）およびデータサイズ（**図 3**）についてはすべてのリクエストを分析対象とした．

まず，図 2 に，ページデータのリクエスト回数の分布を示す．平均リクエスト回数は約 3.5 回であり，2 回以上リクエストされるページデータは約 30%であった．また，約 70%がリクエスト回数が 1 回である一方，最高 18,793 回の場合もあった．次に，データサイズの分布を図 3 に示す．平均データサイズは約 9 Kbyte であり，その分布は 1 Kbyte 以下が約 50%であった．また，約 80%が 5 Kbyte 以下であり，10 Kbyte 以下では約 90%であった．さらに，100 Kbyte 以上は 0.5%であり，最大 50 Mbyte の場合もあった．次に，サービス応答時間の分布を図 4 に示す．サービス応答時間の

☆ サービス応答時間を改善する他の試みとしては先読み型キャッシュなどがあるが，本論文ではキャッシュ置き換えアルゴリズムに改善という立場から新たな手法を提案する．

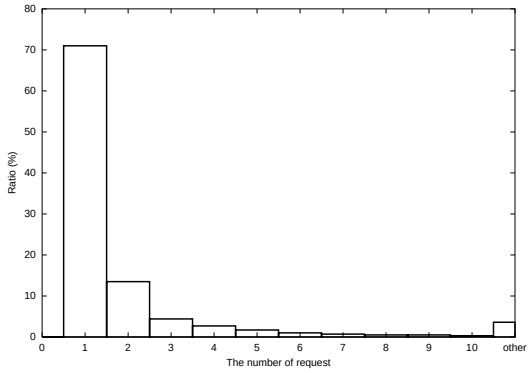

図 2 リクエスト回数の割合
Fig. 2 A ratio of request number.

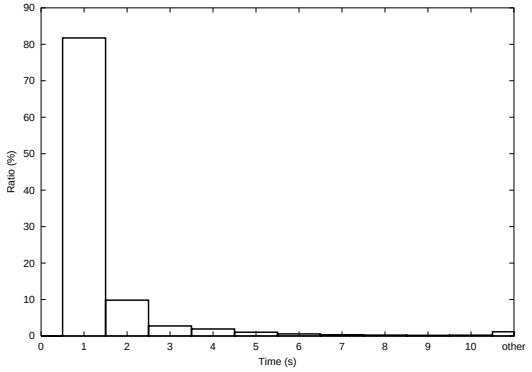

図 4 サービス応答時間の割合
Fig. 4 A ratio of service response time.

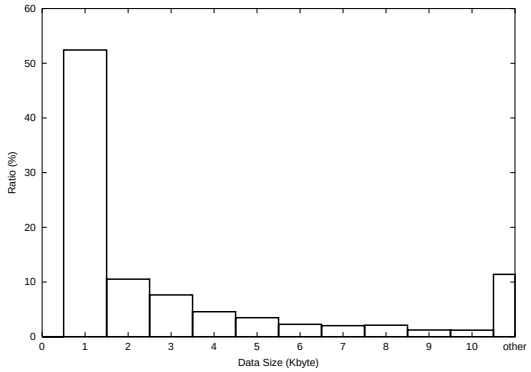

図 3 データサイズの割合
Fig. 3 A ratio of data size.

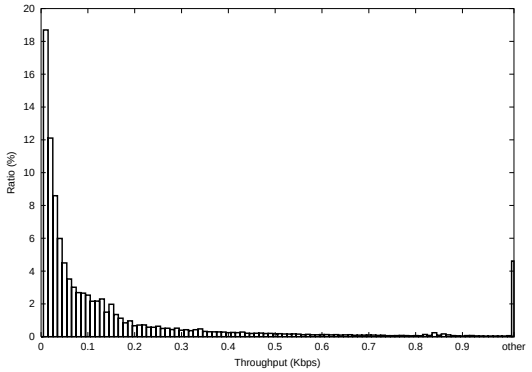

図 5 スループットの割合
Fig. 5 A ratio of throughput.

平均は約 1 秒であり，その分布は 1 秒以内が約 80%であった．また，2 秒以内では約 90%であった．さらに，サービス応答時間が 10 秒以上は約 1%であり，最高約 30 分の場合もあった．最後に，スループットの分布を図 5 に示す．平均スループットは約 243 Kbps であり，その分布は 0.2 Kbps 以下が約 80%であった．また，1 Kbps 以上では 5%であり，最大 46 Mbps の場合もあった．そのほか，WWW アクセスの特徴は文献 7) でも同様に述べられている．

ここで，スループットをより高くし，サービス応答時間をより短くすることを提案手法における改善目標とすると，以上の結果は，次のようにまとめることができる．すなわち，スループットにおいて，全体の約 5%以下である 1 Kbps 以上のページデータによって平均スループットが大きく引き上げられている．このことから，スループットが低いページデータを優先的にキャッシュすることによって，スループットをさらに改善することが期待できる．また，スループットは単位サービス応答時間あたりの転送データサイズで表されるため，同一サイズのページデータの場合，スループットの改善によりサービス応答時間の短縮も期待できる．また，リクエストの約 70%を占めるリクエスト回数が 1 回のページデータを長期間キャッシュすることは限りあるキャッシュ領域を考えれば無駄である．そこで，アクセス頻度の高いページデータを優先的にキャッシュすることによってキャッシュされるページデータの再利用価値を高めることができる．さらに，同一スループットの場合，データサイズの大きなページは取得時間も大きくなることから，データサイズの大きなページを優先的にキャッシュすることによってサービス応答時間の改善に影響すると考えられる．

以上の分析結果から，我々はサービス応答時間を改善するために，アクセス頻度およびデータサイズに加えてスループットを考慮したキャッシュ置き換えアルゴリズムを提案する．

3. 提案手法：GTSFD（Greedy-Triple，Size Frequency Distance）

3.1 概　　要

従来のキャッシュ置き換えアルゴリズムでは，アクセ

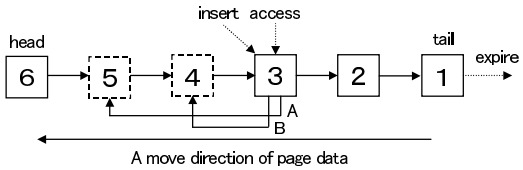

図 6 提案手法
Fig. 6 A proposal method.

ス頻度とデータサイズをキャッシュ優先度を決定する評価指標として用いることで，ヒット率やバイトヒット率の改善を目的とした手法[1),2)]が一般的である．しかし，実際は同一データサイズであっても，WWW サーバごとに経路途中に存在するルータ数やリンク帯域が異なり，ページデータの取得時間も異なる．このことは，ヒット率やバイトヒット率を優先する従来手法がサービス応答時間を明示的に改善するものではないことを意味している．例を示すと，同一サイズでデータ取得時間が異なるページデータどうしがアクセス頻度を基準に対等に扱われるため，アクセス頻度が低くデータ取得時間が大きいページデータが，アクセス頻度が高くデータ取得時間が小さいページデータのためにキャッシュから追い出され，結果的にサービス応答時間の低下を招く場合がある．

そこで本論文では，従来のキャッシュ置き換えアルゴリズムで用いられていたアクセス頻度とデータサイズに加えて，スループットをキャッシュ優先度を決定する評価指標として採用し，サービス応答時間の改善に重点をおいた新たなキャッシュ置き換えアルゴリズム（GTSFD：Greedy-Triple, Size Frequency Distance）を提案する．スループットを考慮する根拠は，2 章で述べたとおり単位サービス時間あたりの転送データサイズで表されるスループットを改善することにより，サービス応答時間の改善を期待できるためである．

GTSFD の動作概要は次のとおりである．すなわち，**図 6** においてアクセス頻度およびデータサイズが大きく，スループットが小さいページデータほどキャッシュ優先度（ページデータをより長期間キャッシュするための評価値）を高くし (A)，その逆の場合は低くする (B)．また，1 度キャッシュしたページデータが再びリクエストされた場合は，アクセス頻度が高いページデータほどキャッシュ優先度を高くし (A)，その逆の場合は低くする (B)．一方，アクセス頻度，データサイズ，スループットは各々次元（単位）が異なるため，そのままでは対等な価値として評価することはできない．そこで，各評価値のスケーリングを行い，対等な価値として扱えるようにする．

表 1 相関
Table 1 Correlation.

区間	相関
アクセス頻度－データサイズ	−0.00062
データサイズ－ 1/スループット	−0.00036
1/スループット－アクセス頻度	0.00046

以下では，アクセス頻度，データサイズ，スループットの 3 つの評価指標からキャッシュ優先度を決定する評価式を導出し，WWW キャッシュサーバへの実装および評価を行う．

3.2 優先度評価式の導出

従来のキャッシュ置き換えアルゴリズムでは，キャッシュ優先度を決定する評価指標としてアクセス頻度やデータサイズを用いるのが一般的であった．提案手法ではこれらに加えて，スループットを新たな評価指標として採用する．すなわち，スループットを（転送データサイズ/サービス応答時間）と定義し，その逆数，つまり，単位データサイズあたりのサービス応答時間が長いほどキャッシュ優先度を高くする．さらに，ここでは，ページデータのアクセス頻度，データサイズ，1/スループットの 3 つの評価指標を対等に評価することを提案する．このことは，前述に加え，アクセス頻度が多いほど，また，データサイズが大きいほどキャッシュ優先度を高めることを意味する．

ここで，この 3 つの評価指標の各々が，他に依存する場合は独立な評価指標になりえない．そこで，まず，各々がそれぞれ独立であることを確認するため，評価指標どうし（アクセス頻度，データサイズ），（データサイズ，1/スループット），（1/スループット，アクセス頻度）の相関を求めた（**表 1**）．

表 1 の結果から，各々の相関がほぼ 0 であることから，これらの評価指標は互いに独立と見なせる．提案手法では，これら 3 つの評価指標で張られる空間を 3 次元の直交空間（ユークリッド空間）と定義し，原点からの距離（ユークリッド距離）を該当ページデータのキャッシュ優先度とする．ここで，それぞれの評価指標は次元（単位）が異なるため評価指標どうしに対等な価値を与えるスケーリングを行う必要がある．また，各々の出現分布（分布関数：ヒストグラム）は，特定の分布関数に従うと仮定することは困難であるため，それらの統計量（平均や分散など）を用いた近似分布関数を与えることはできない．そこで，各々の評価指標の平均値を基に指数関数的な価値に基づく距離を導入する．以上のことを，データサイズを例にとって以下で説明する．

まず，データサイズを α，データサイズの平均を $\overline{\alpha}$

として，原点からの距離（データサイズ評価値：α_{val}）を次式で与える．

$$\text{データサイズ：} \quad \alpha_{val} = 1 - exp\left(-\frac{\alpha}{\overline{\alpha}}\right) \quad (1)$$

この関数は，一般に一次遅れ系の応答関数と呼ばれ，α が小さいほど原点からの距離が短く，逆に，α が大きいほど原点からの距離が長くなる．ここでは，$\overline{\alpha}$ を時定数と見なしており，$\alpha = \overline{\alpha}$ であるならば原点からの距離（つまりデータサイズ評価値）が 0.632 となる．すなわち，平均値による正規化操作を行っていることを意味する．さらに，他の 2 つの評価指標（アクセス頻度評価値：β_{val}，1/スループット評価値：γ_{val}）についても同様に，

$$\text{アクセス頻度：} \quad \beta_{val} = 1 - exp\left(-\frac{\beta}{\overline{\beta}}\right) \quad (2)$$

$$\text{1/スループット：} \quad \gamma_{val} = 1 - exp\left(-\frac{\gamma}{\overline{\gamma}}\right) \quad (3)$$

と表すことができる．式 (1)〜式 (3) より，3.2 節で述べた提案手法（GTSFD）におけるキャッシュ優先度（$GTSFD_{val}$）は 3 変数のユークリッドノルムを用いて次式で定義する（ここで，分母 $\sqrt{3}$ はキャッシュ優先度を正規化するための係数である）．

キャッシュ優先度：

$$GTSFD_{val} = \sqrt{\frac{\alpha_{val}^2 + \beta_{val}^2 + \gamma_{val}^2}{3}} \quad (4)$$

次に，式 (4) で表されるキャッシュ優先度では，ページデータの要求ごとに各変数の平均値を更新していかなければならない．そこで，平均の逐次式を求める．

まず，ページデータの集合 $\{x_1, x_2, \cdots, x_k, \cdots, x_N\}$ が与えられたとき，平均データサイズを $\overline{\alpha}_{x_k}$ とすると，

$$\text{平均：} \overline{\alpha}_{x_k} = \frac{1}{k}\sum_{i=1}^{k} \alpha_{x_i} \ (k = \text{データ数}) \quad (5)$$

と表すことができる．

ここで，$k-1$ と k の関係を求める．

$$\overline{\alpha}_{x_{k-1}} = \frac{1}{k-1}\sum_{i=1}^{k-1} \alpha_{x_i} \quad (6)$$

また，式 (5) より，$\overline{\alpha}_{x_k}$ は，

$$\overline{\alpha}_{x_k} = \frac{1}{k}\left(\sum_{i=1}^{k-1} \alpha_{x_i} + \alpha_{x_k}\right) \quad (7)$$

式 (6) より，

$$\sum_{i=1}^{k-1} \alpha_{x_i} = (k-1)\overline{\alpha}_{x_{k-1}} \quad (8)$$

表 2 Squid の構成
Table 2 Configuration of Squid.

name	value
Version	Squid-2.4STABLE7
Extension	GTSFDv1.0p24
Cache memory	512 MB
Cache disk	4.8 GB

表 3 サーバの構成
Table 3 Configuration of server.

name	value
Model	PC/AT Compatible
OS	Slackware 8.0
Kernel	Linux 2.4.19
Physical memory	2 GB
Storage Disk	72 GB*4 (SCSI)
Network I/F	100 Mbps

表 4 ネットワークの構成
Table 4 Configuration of network.

name	value
Network name	RAINS (Class-B Network)
Number of users	8,600
Number of clients	4,200
WAN bandwidth	15 Mbps

式 (8) を式 (7) に代入すると，

$$\overline{\alpha}_{x_k} = \frac{1}{k}\{(k-1)\overline{\alpha}_{x_{k-1}} + \alpha_{x_k}\} \quad (9)$$

ここで，k が十分大きい場合は，以下のような近似が可能である．

$$\overline{\alpha}_{x_k} = \overline{\alpha}_{x_{k-1}} + \frac{1}{k}\alpha_{x_k} \quad (10)$$

さらに，平均リクエスト回数，平均スループットについても同様に求まる．以上により，キャッシュ優先度を決定する評価式（式 (4)）および各評価値の平均を求める逐次式（式 (9)，式 (10)）が求められた．

3.3 評価実験

提案手法は，WWW キャッシュサーバとして広く利用されている Squid [8] のソースコードを変更することで実装を行った（**表 2**）．提案手法を実装した WWW キャッシュサーバ（Squid）は，**表 3** に示す計算機上で運用し，比較のため LRU（Least Recently Used），LFUDA（Least Frequently Used with Dynamic Aging），GDSF（Greedy-Dual Size-Frequenccy）の 3 つの従来手法によるデータ取得も行った．また，各手法におけるデータ取得ではキャッシュ置き換えアルゴリズムを変更した以外の設定は同一とした．さらに，WWW リクエストは**表 4** に示すネットワークにおいて，ゲートウェイルータでポートリダイレクトする透

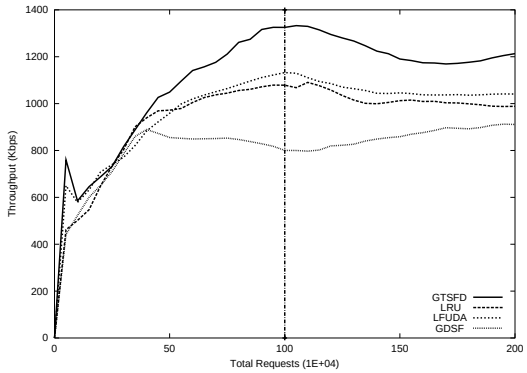

図 7 スループット
Fig. 7 Throughput.

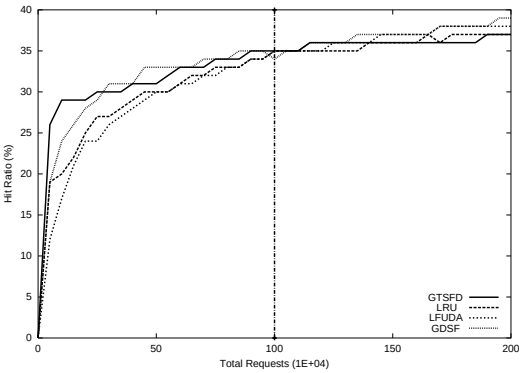

図 9 ヒット率
Fig. 9 Hit ratio.

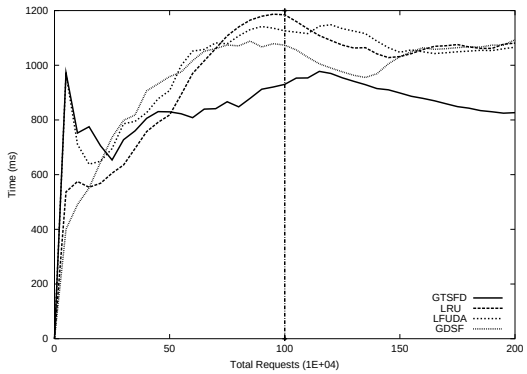

図 8 サービス応答時間
Fig. 8 Service response time.

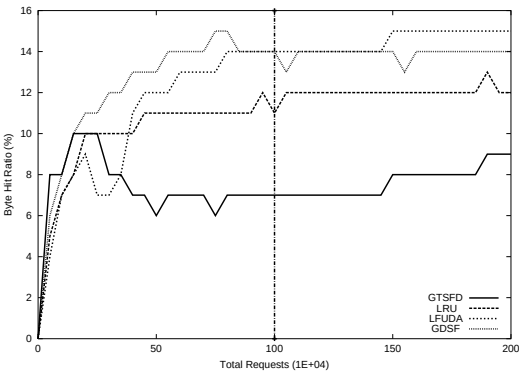

図 10 バイトヒット率
Fig. 10 Byte hit ratio.

過プロキシによって収集し，各手法のデータはメモリキャッシュおよびディスクキャッシュ領域を初期化した状態から 200 万リクエスト取得した．WWW キャッシュサーバを運用したネットワークは，バックボーン帯域が 15 Mbps，接続端末数が約 4,200 台，ネットワーク利用者数が約 8,600 人のネットワークであり，データ取得期間中の 1 日あたりの WWW キャッシュサーバへのリクエスト数は約 100 万リクエストであった．また，各手法のデータ取得において，100 ± 5 万リクエストの範囲でディスクキャッシュ溢れを確認した．

3.4 評価・考察

まず，スループットでは，200 万リクエスト時点で GTSFD が最も高く従来手法に比べて約 20%以上改善されている（**図 7**）．また，サービス応答時間についても同様に，GTSFD が最も短く従来手法に比べて約 25%以上改善されている（**図 8**）．さらに，ヒット率では，従来手法および GTSFD ともにほぼ同等の結果（ヒット率：37〜39%）であった（**図 9**）．また，バイトヒット率では，GTSFD が最も低い結果となった（**図 10**）．

ここで，サービス応答時間が従来手法に比べて約 25%以上改善された要因は次のように考察することができる．すなわち，スループットにおいて従来手法に比べて約 20%以上改善されているため，単位データサイズあたりのデータ転送時間が短縮された効果によるものである．このことから，スループットを改善することによってサービス応答時間の改善するという提案手法の目的が明確に実験結果に反映されているといえる．サービス応答時間は，利用者側から見た WWW サービス使用感に直接影響するものであり，この点からも本実験結果は GTSFD の有効性を示している．

次節では，提案手法における 3 つの評価指標の評価値に重み付けを行うことによって，キャッシュ性能の変化を検証する追加実験について述べる．

3.5 追 加 実 験

3.3 節における評価実験によって，提案手法はサービス応答時間が改善されていることを確認したが，バイトヒット率が低い結果になった．この原因は，次のように考察することができる．すなわち，サービス応答時間が改善されていることから，3.2 節におけるデータ

表 5 重み付けパターン
Table 5 A pattern of weight.

パターン	α_{val}: β_{val} : γ_{val}
1	1 : 1 : 1
2	5 : 1 : 1
3	1 : 5 : 1
4	1 : 1 : 5

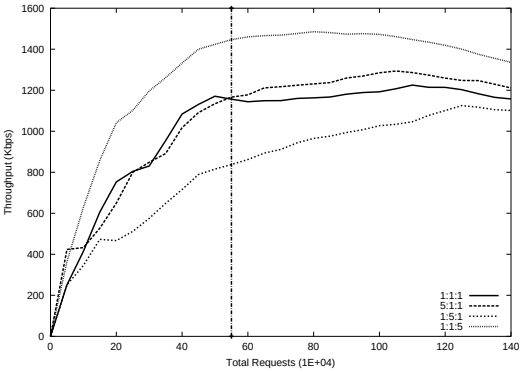

図 11 スループット
Fig. 11 Throughput.

サイズ評価値（α_{val}）によってデータサイズがより大きいページデータの評価値が高くなる一方，1/スループット評価値（γ_{val}）によって，サービス応答時間がより長く，データサイズがより小さいページデータの評価値が高くなる．つまり，キャッシュ優先度の評価式（式 (4)）において，1/スループット評価値とデータサイズ評価値とで，データサイズの影響が相殺され，結果的にキャッシュ優先度に与えるデータサイズの影響が小さくなる．このため，データサイズが大きく影響するバイトヒット率の結果が低くなっている．

そこで，次のような追加実験を行った．すなわち，3.2 節において導出したキャッシュ優先度の評価式（式 (4)）において，3 つの評価指標（アクセス頻度，データサイズ，スループット）の評価値（α_{val}，β_{val}，γ_{val}）に**表 5** に示す重み付けを行い，各パターンにおける実験結果から特徴分析を行った．本実験の目的は，各評価指標がキャッシュ優先度に与える影響力を変更することによって，提案手法のキャッシュ性能の変化を検証することである．ここで，実験は表 2 におけるディスクキャッシュ領域を 2.4 GB に変更し，データは 140 万リクエスト取得した．その他の条件は 3.3 節で行った実験と同様とした．また，各手法のデータ取得において，55 ± 5 万リクエストの範囲でディスクキャッシュ溢れを確認した．

3.6 評価・考察

まず，140 万リクエスト時点でのスループットでは，

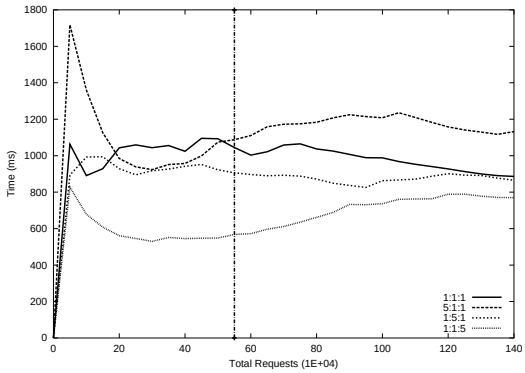

図 12 サービス応答時間
Fig. 12 Serveice response time.

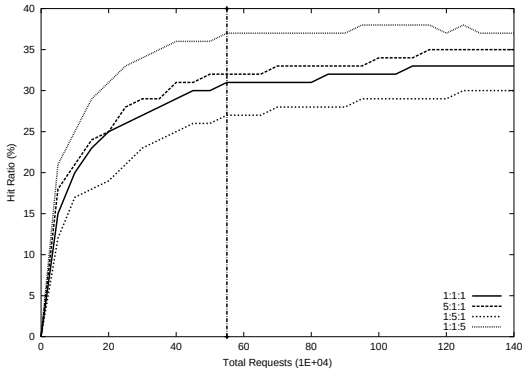

図 13 ヒット率
Fig. 13 Hit ratio.

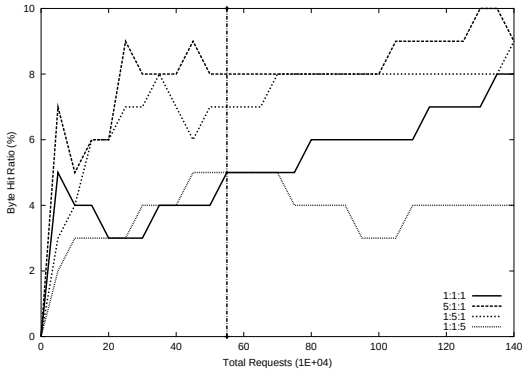

図 14 バイトヒット率
Fig. 14 Byte hit ratio.

スループット評価値を高く重み付けしたパターン 4 が最も高く，次いでアクセス頻度評価値を高く重み付けしたパターン 2 となり，データサイズ評価値を高く重み付けしたパターン 3 が最も低い結果となった（**図 11**）．また，サービス応答時間では，パターン 4 が最も短く，次いでパターン 3 となり，パターン 2 が最も長い結果となった（**図 12**）．さらに，ヒット率で

は，パターン 4 が最も高く，次いでパターン 2 となり，パターン 3 が最も低い結果となった（**図 13**）．最後に，バイトヒット率では，パターン 2 およびパターン 3 が最も高く，パターン 4 が最も低い結果となった（**図 14**）．

以上の実験結果から次のように考察することができる．すなわち，パターン 4 では 1/スループット評価値においてデータサイズがより小さいページが優先的にキャッシュされるため，キャッシュされるページデータの相対的な数が多くなることからヒット率が改善され，パターン 2 ではアクセス頻度評価値の重みを高くすることによって，再リクエストされる可能性が高いページデータを優先的にキャッシュすることからヒット率が改善している．また，パターン 3 でデータサイズ評価値の重みを高くすることによって，より大きなデータサイズのページを優先的にキャッシュすることからバイトヒット率が改善している．さらに，パターン 4 でスループット評価値の重みを高くすることによって，スループットが改善しているが，3.3 節での実験結果と同様に，1/スループット評価値においてデータサイズがより小さく，サービス応答時間がより長いページが優先的にキャッシュされるため，バイトヒット率が低い結果となった．

以上のことより，評価値の重み付けによって，重み付けパターンに応じた性能差が現れていることを確認した．特に，データサイズの評価値の優先度を大きくした場合，3.3 節の実験で性能が振るわなかったバイトヒット率を改善することができた．このことから，提案手法は運用条件やキャッシュ性能改善の目的によって，キャッシュ性能を調整可能であり，ヒット率，バイトヒット率，スループット，サービス応答時間において複合的に性能改善が実現できる手法であるとまとめることができる．

4. ま と め

4.1 今後の課題

本論文では，従来のキャッシュ置き換えアルゴリズムでキャッシュ優先度を決定する評価指標として用いられていたアクセス頻度およびデータサイズに加えて，スループットを考慮することによってサービス応答時間を改善するキャッシュ置き換えアルゴリズムを提案し，その有効性を評価実験により確認した．しかし，さらに次のような検討すべき課題があると考えられる．

まず，提案手法ではキャッシュ優先度の評価値としてスループットを採用しているが，スループットはリクエストのあった時間帯のネットワークやサーバの負荷状態によって変動していると考えられる．このため，スループットを公平な評価指標として用いるには，負荷変動を考慮して動的に評価値の重み付けを変更するための拡張が必要であると考えられる．

また，提案手法は単体の WWW キャッシュサーバのサービス応答時間の改善を実現したものであるが，WWW プロキシサーバのクラスタリング[9]や多段プロキシシステム[10)~12)]での性能改善効果を検証することが考えられる．これらのマルチサーバシステムでは，提案手法による単体サーバの性能改善によってシステム全体への性能改善効果があると考えられる．

最後に，本論文における提案手法の有効性をさらに明確に確認するため，引き続き運用データ取得を行っており，以上の課題についても引き続き検討していく．

4.2 結　　論

本論文では，まず，WWW キャッシュサーバのアクセスログから WWW アクセスの特徴分析を行い，サービス応答時間を改善するキャッシュ置き換えアルゴリズムの評価指標としてスループットを採用した．次に，アクセス頻度，データサイズ，スループットをキャッシュ優先度を決定する評価指標とし，指数関数的な価値に基づく評価式を導出して実システム（Squid）への実装を行った．これを実ネットワーク上で運用し，その運用データから評価を行った結果，従来手法と比べておおむね 20%以上のサービス応答時間の改善を確認した．さらに，提案手法におけるキャッシュ優先度の評価式において，3 つの評価指標に各々重み付けを行った場合の追加実験を行った結果，各重み付けパターンごとに性能改善の特徴が見られた．特に，データサイズを優先的に重み付けすることによって，重み付けを行わない場合に比べて，提案手法でのスループット改善によるバイトヒット率の性能低下を軽減することができた．

以上の結果から，提案手法はサービス応答時間の改善を実現し，さらに，3 つの評価指標の重み付けを変更することによって，WWW キャッシュサーバの運用形態や性能改善の目的に応じたキャッシュ性能の調整が可能な手法であり，すべてのキャッシュ性能を複合的に改善可能な手法であると結論することができる．

謝辞　本研究の遂行にあたっては，琉球大学総合情報処理センターの方々に多くの協力をいただいた．また，本研究は，文部科学省および日本学術振興会の科学研究費補助金（基盤 B：課題番号 13450165）の助成により行われた．ここに深く感謝いたします．

参 考 文 献

1) Abrams, M., Standridge, C.R., Abdulla, G., Williams, S. and Fox, E.A.: Caching Proxies: Limitations and Potentials, *4th International World Wide Web Conference* (1995).
2) 大澤範高，早野文孝，弓場敏嗣，箱崎勝也：WWW プロキシサーバのログに基づいたキャッシュ置き換えアルゴリズムの評価，情報処理学会マルチメディアと分散処理研究会報告，Vol.96-DPS-74, No.33, pp.191–196 (1996).
3) Scheuermann, P., Shim, J. and Vingralek, R.: A Cache for Delay-Conscious Caching of Web Documents, *6th International World Wide Web Conference*, pp.725–734 (1997).
4) Wooster, R.P. and Abrams, M.: Proxy Caching that Estimates Page load Delays, *6th International World Wide Web Conference*, pp.325–334 (1997).
5) Pei, C. and Sandy, I.: Cost-Aware WWW Proxy Caching Algorithms, *Proc. 1997 Usenix Symposium on Internet Technologies and Systems* (1997).
6) Dilley, J., Arlitt, M. and Perret, S.: Enhancement and Validation of Squid's Cache Replacement Policy, HP Labs Technical Reports, HPL-1999-69 (1999).
7) 西川記史，細川貴史，森 靖英，吉田健一，辻洋：WWW トラフィック特性に基づくキャッシュ方式の提案，情報処理学会論文誌，Vol.40, No.12, pp.4333–4343 (1999).
8) Squid. http://www.squid-cache.org/
9) 大須賀圭人，河合栄治，知念賢一，山口 英：プロキシサーバの故障を考慮した HR キャッシュクラスタの性能解析，情報処理学会マルチメディアと分散処理研究会報告，Vol.99-DPS-15, No.3, pp.13–18 (1999).
10) 井上博之，坂本岳史，山口 英，尾家祐二：分散 WWW キャッシュシステムの構成自動設定機構，情報処理学会論文誌，Vol.40, No.12, pp.4333–4343 (1999).
11) 田中友英，篠田陽一：WWW における動的経路制御を用いた多段キャッシュシステム，情報処理学会マルチメディアと分散処理研究会報告，Vol.97-DPS-83, No.9, pp.49–54 (1997).
12) 酒井明広，知念賢一，砂原秀樹，湊小太郎：分担型キャッシュシステムの設計と実装，情報処理学会マルチメディアと分散処理研究会報告，Vol.98-DPS-87, No.30, pp.173–178 (1998).

(平成 14 年 8 月 26 日受付)
(平成 15 年 2 月 4 日採録)

長田 智和（学生会員）

昭和 49 年生．平成 10 年琉球大学工学部情報工学科卒業．平成 12 年同大学院理工学研究科情報工学専攻博士前期課程修了．現在，同大学院理工学研究科総合知能工学専攻博士後期課程在学中．ネットワークアーキテクチャ，ネットワーク運用管理，通信プロトコルに関する研究に従事．

神里 大

昭和 51 年生．平成 13 年琉球大学教育学部総合科学課程情報教育コース卒業．現在，同大学院理工学研究科情報工学専攻博士前期課程在学中．WWW キャッシュシステム，教育教材開発に関する研究に従事．

谷口 祐治（正会員）

昭和 31 年生．昭和 54 年琉球大学理工学部電気工学科卒業．昭和 55 年同大学工学部研究課程修了（研究生）．昭和 56 年同大学工学部電子・情報工学科教務技官．昭和 63 年同大学短期大学部電気工学科助手．平成 5 年同大学工学部情報工学科講師．平成 10 年同大学総合情報処理センター講師，現在に至る．現在，ネットワークアーキテクチャ，情報教育に関する研究に従事．日本教育工学会，電子情報通信学会，IEEE 各会員．

玉城 史朗（正会員）

昭和 30 年生．昭和 54 年琉球大学理工学部機械工学科卒業．昭和 56 年徳島大学大学院工学研究科情報工学専攻修了．昭和 58 年大阪大学大学院基礎工学研究科物理系退学．同年同大学基礎工学部助手．昭和 61 年岡山理科大学理学部応用数学科講師．平成元年琉球大学工学部機械工学科助教授．平成 8 年琉球大学工学部情報工学科教授，現在に至る．現在，環境問題と自然エネルギー応用，最適化理論の研究に従事．工学博士．電子情報通信学会，日本太陽エネルギー学会，日本工学教育協会各会員．