

WWW キャッシングサーバにおける置換 アルゴリズム動的選択法の提案・検討

上 田 武 史[†], 中 山 雅 哉^{††}

近年の WWW では、コンテンツの多様化、大容量化が急速に進み、コンテンツ利用状況は変動が大きくなりつつある。本稿では、通常単一使用される WWW キャッシングにおける置換アルゴリズムを複数用い、特徴に従い、アクセス状況に応じて動的に選択することを提案する。本研究では、workload を用いたシミュレーション評価を行い、提案手法について検討した。

Dynamic Selection of Replacement Policy on WWW Cache

TAKEFUMI UEDA[†] and MASAYA NAKAYAMA^{††}

Contents has been become more multifaceted rapidly on today's WWW. Large fluctuations of contents utilization has been seen. To follow these dynamic utilization, dynamic selection of replacement policy is proposed in this study, besides of use of only one replacement policy. Simulations with some workloads were run to evaluate this method.

1. はじめに

インターネット、特に World Wide Web(WWW) において、キャッシング技術により一度取得されたコンテンツについて、再度同じコンテンツを取得する必要性を無くすということが広く行われている。キャッシュサーバ上で、より有効にネットワーク、ディスク、メモリ等の資源を活用できるように、どのオブジェクトをキャッシュから破棄するかを決定する、置換アルゴリズムの研究も広くなされてきた。これらのアルゴリズムは、過去のアクセス統計などに基づき、キャッシュを設定する段階において目的に最も有効であると期待されるものが選択される。近年に至るまでのインターネットにおいては、テキスト、静止画像が支配的であり、トラフィックの特性も比較的経験的に得られた法則に収束される結果が見られることが多かった。従って、得られる統計情報に基づき、適した置換アルゴリズムを決定し運用することは有効な手段であった。しかし、近年においては、大サイズイメージファイル、文書アーカイブ、動画アーカイブ、オーディオファイル等様々なコンテンツが WWW サービスを通してやりとりされるようになり、以前より複雑性、多様性が

増している。この結果、従来よりも、時間的変動、アクセス状況等は複雑になってきており、WWW キャッシュサーバも、より変動に柔軟に対処できることが望まれる。キャッシュの評価軸として、ヒット率、バイトヒット率といったものが挙げられるが、現在における多様化したコンテンツ、とくにコンテンツのサイズに注目したときに、アクセス変動によっては、単一アルゴリズムでは偏った特徴に対応した置換を行うため、基本的に両方を共に上げることは難しい。そこで、本稿では、適宜に利用状況にあわせてキャッシュサーバが動的に置換アルゴリズムを選択することで、変動するアクセスに追従できないか、ということを検討した。以下まず第 2 章で関連事項に関して記述し、続いて 3 章において提案手法の概要と注目点を記述する。4 章でシミュレーションによる評価結果を示し、5 章にまとめと今後の課題を記述する。

2. 現在での WWW キャッシングの動向とその改善

現在においては、ISP(Internet Service Provider) のほとんどがキャッシュを設置しているといわれており、その需要は非常に高い。リソースの有効利用のために、WWW キャッシングは有効な技術として現在では広く普及している。

[†] 東京大学
現在、日本ヒューレット・パカード株式会社
^{††} 東京大学情報基盤センター

2.1 広域分散キャッシング

キャッシング技術の方向性として、現在では、Akamai¹⁾に代表される広域分散キャッシングに注目が集まっており、現在トレンドとなっている。近年では Inktomi²⁾、Cache Flow³⁾ などキャッシュサーバそのものの技術に力をいれていたベンダーも Akamai と提携し、配送 + キャッシュの形が主流になりつつある。この方式では、世界規模でコンテンツをキャッシュするサーバを大量に配置し、WWW サーバからコンテンツをキャッシュへコピーしておくことにより、WWW サーバ自体へのアクセスを減らしている。この方式ではクライアントのリクエストが一番近い、もしくは経路情報などにより最適なキャッシュサーバからコンテンツを取得するため、クライアントは遠隔地の WWW サーバそのものにアクセスする必要がない。関連したキャッシング戦略として、コンテンツ更新方式に関する研究⁴⁾、キャッシュサーバの配置に関する研究等⁵⁾ がなされてきた。

2.2 単体キャッシュの改善

前述のようなコンテンツ配信方式では、クライアントリクエストのトラフィックを軽減する事はできるが、コンテンツ更新時などのデータのやりとりに対するトラフィックが生じる。また、サーバを複数化すること、世界規模で配置する事は高いコストを必要とし、誰もができる解決方法ではない。

もう一つのキャッシュの効率化を目指すアプローチの方向は、単体のサーバの能力を向上させることである。単体のサーバをソフトの面で改善できるならば、必要コストが削減でき、また一点のみを管理することは配送の様に多くの管理労力を必要としない。このような点でのアプローチとして従来より研究が盛んになされてきたのが、置換アルゴリズムの研究である。

2.3 関連した研究：置換アルゴリズム

WWW キャッシュで用いられる置換アルゴリズムとしては、非常に多くの研究がなされてきた。置換アルゴリズムを決定する上で、重要になってくるのは、オブジェクトへのアクセスのうち何をパラメータとして用いるか、である。近年では様々なパラメータの用い方が考えられているが、そのうち、LRU のようなオブジェクトの参照時間、LFU のような参照頻度履歴、Size アルゴリズムのようなオブジェクトのサイズをパラメータとして考慮したアルゴリズムが代表的である。

2.3.1 置換アルゴリズムとサイズパラメータ

キャッシュサーバでの置換アルゴリズムは通常リクエストのアクセス統計などに基づき、そのキャッシュ

サーバにおいて観察されるアクセスパターンに対し効果的な置換を行うことが期待されるアルゴリズムがキャッシュの管理者によって選択されるのが一般的である。統計的手法には過去の傾向のデータの蓄積が必要であるが、アクセスする集団が変動した場合など、アクセスパターンも変動することが予測され、また、現在よく参照されるオブジェクトのセットが存在していたとしても、これらのオブジェクトセットは新たなオブジェクトセットの出現により容易に変化し得る。こうした状況では、リソースキャパシティを効率よく運用するために、管理者はその変化に敏感であることが要求される。つまり、管理者にはなんらかの形で定期的な監視、分析が強いられ、再び変化に対応した選択、構築をしなければならない。

またコンテンツサイズに注目した場合、サイズを優先付に用い、大サイズのオブジェクトを破棄することでヒット率を高める、というアルゴリズムがある。これらは、大サイズオブジェクトの再アクセスに対し不利であり、バイトヒット率は高くならない。また、サイズを優先付に用いない場合は、バイトヒット率は高くなるが、無駄な大サイズオブジェクトを蓄積する可能性が高く、ヒット率が低くなる。過去から現在のインターネットでは、テキスト、HTML が支配的であり、そのためサイズ分布には、大きな局所性がみられた。しかし、現在から将来のインターネットでは、これらの局所性は崩れていき、次第に長期に渡る統計データによる手法では、短期変動に対し追従しきれない状況になっていくことが予想される⁶⁾。

2.3.2 サイズ別アクセス動向

LAN 環境において、時間的にみると、利用目的に従って必要とする外部データが比較的小サイズのものである時間帯と、比較的大サイズのデータが必要とされる時間帯が発生する、といった状況が生じることが予測される。このように多様なデータが出入りする LAN 環境での proxy サーバへのアクセスをサイズという視点よりみたときに、アクセス傾向には時間的な偏りがある、と考えられる。図 1 は実アクセスのログから抜きだした 1 時間分のアクセス履歴の、サイズが 50k バイト以上のオブジェクトへの参照率である。2001 年 1 月 15 日の 13 時から 14 時迄と、同日 16 時から 17 時までの記録である。尚サイトの利用制限等のため、取得されたコンテンツの平均サイズは 6.3k バイトとやや小さい。図 1 のように時間的に取得されるコンテンツサイズの変動は観察されるが、同じものへの参照が見られるかどうか重要である。図 2 は、図 1 に示した変動に対し、50k バイト以上のサイズをも

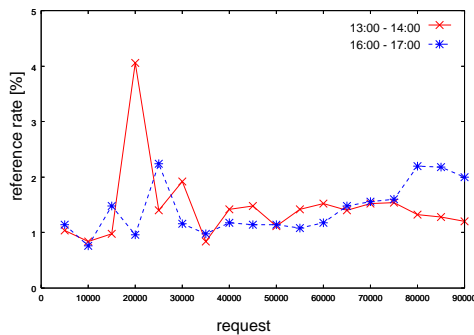


図 1 50k バイト以上のオブジェクトへの期間 (5000 リクエスト) 内参照率: 東京大学 ECC サイト 2001 年 1 月 15 日 13-14 時、16-17 時

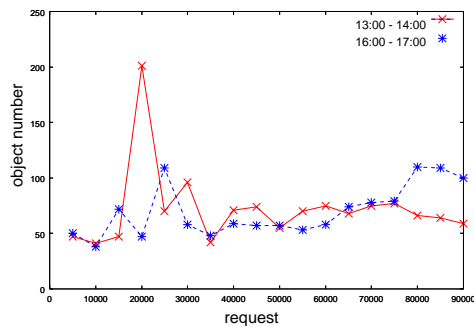


図 2 再参照された 50k バイト以上のサイズのオブジェクト数

つオブジェクトのうち、2 度以上参照されたオブジェクトの個数の変動を調査した結果である。この結果では、サイズ参照率と再利用されるオブジェクト数には大きな関連性が見られる。尚この期間では、期間毎の参照数 (参照頻度を期間毎に計測) と、始めのリクエストからを考慮した場合の参照数 (参照頻度履歴を保持) は殆んど同じであり、期間的な局所性があることもわかった。

3. 置換アルゴリズム選択モデル

前章で触れたように、例えばある LAN 環境において、利用目的に応じて必要とする外部データが比較的小サイズのもの (低解像度画像、小サイズデータ等) である時間帯と、比較的大サイズのデータ (高解像度画像、プログラム、映像アーカイブ等) が必要とされた時間帯が生じることを想定する。以前と比べ大サイズコンテンツが著しく増大すると、参照回数が多い大サイズオブジェクトも多く存在するようになるということが考えられるため、このような利用状況は十分に考えられる。

単一の置換アルゴリズムで運用がなされている場合は、時間変動に対し前章で記述したトレードオフに

より、置換効率が悪くなる期間が生じ、キャッシュの無駄利用率が増えかねない。このような利用に適したキャッシングとしては、

- 大サイズオブジェクトが多くアクセスされるようならば、大サイズオブジェクトをキャッシュすることで、よく参照される大サイズオブジェクトをキープする
- 小さいオブジェクトが多くアクセスされるようならば、大サイズオブジェクトを破棄することでスペースを作り、よく参照される小サイズオブジェクトをキープする

といったようなキャッシングが望ましい。しかし大サイズオブジェクトをキャッシュしないようにするアルゴリズムと、サイズに関わらずキャッシュするアルゴリズムはサイズの点で見れば本質的に異なるため、このようなキャッシングはできない。こうした場合は、現在では、状態を監視して、適する状態に設定し直す、といったことが考えられる。しかし、それを人手によって行うのは、変化が多い場合は効率が悪い。

そこで、現在では静的に決定されているキャッシュ上の置換アルゴリズムを、複数用意し、アクセス情報に基づき動的に選択し、再設定することを提案した。

3.1 選択手法の概要

利用状況に応じて 2 つの系統のアルゴリズムを用いることとした。一方は比較的大サイズオブジェクトへの参照が無いときのためであり、他方は大サイズオブジェクトの利用率が上がったとき用いる。まず、あるサイズをサイズ閾値と決め、その閾値以上のサイズの、期間内参照率を計測する。実際の方法として、各取得プロセスにおいて、サイズ情報の取得をしたときに、閾値以上のオブジェクトであればカウントし、カウントされた割合が過去一定時間のアクセスの内で、パラメータ閾値、即ちある参照率を越えた場合、サイズをパラメータとしないアルゴリズムが良いと判断し、選択する。逆に一定時間内での大サイズのオブジェクトへのアクセス数が少なくなった場合は、サイズパラメータを利用して、不要な大サイズオブジェクトをできるだけ破棄するようにする。このアルゴリズムをコードで表したものが、図 3 である。

3.2 選択アルゴリズム候補と閾値

ここでは様々なアルゴリズムのなかから、サイズパラメータを用いて小さいものを優先的にキャッシュするものと、サイズパラメータを用いないでキャッシュするもの、という単純な、相反するアルゴリズムを選択することとした。そこで、参照頻度をベースとした優先づけをする 2 つのアルゴリズム、即ち、サイズ情

```

requests++;
if(obj_size>=Size_Threshold)
  large_obj_counter++;
if(requests==check_time){
  judge_rate = large_obj_counter/requests;
  if(judge_rate>=Rate_Threshold){
    if(!alg2_used){
      alg2_used = 1;
      alg1_used = 0;
    }
  }else{
    if(!alg1_used){
      alg1_used = 1;
      alg2_used = 0;
    }
  }
  requests = 0;
  large_obj_counter = 0;
  judge_rate = 0.0;
}

```

図 3 選択手法の概要

報を用いて大サイズオブジェクトは優先的に破棄する Size-Frequency(SF) と、サイズ情報に関係なくキャッシュする Least Frequently Used(LFU) を用いることとした。サイズ閾値、即ちカウントするサイズの下限の決定については、決定的な手法はないため、様々な手法が考えられるが、広く行なわれている統計的手法よっての決定を想定し、過去ログの長期に渡る参照コンテンツ分布をみることによって決定する方法を採用する。図 4 は、東京大学 ECC サイトのプロキシサーバに記録されたアクセスログ (2001 年 1 月 15 日から 18 日) 約 434 万件より得られたもので、クライアントに転送されたオブジェクトサイズに対する参照頻度である。

このサイズ分布は、対数正規分布として、 $\mu = 4.00$ 、 $\sigma = 1.06$ で近似された。

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma x} \exp\left(-\frac{(\log(x) - \mu)^2}{2\sigma^2}\right) \quad (1)$$

内訳は、1kB 以下のオブジェクトの参照率は 38.1%、2kB 以下 54.8%、5kB 以下で 73.3% という結果が得られた。得られた傾向に対する有効なキャッシングを検討する際には、この結果に基づいた閾値の決定が有効であると思われる。従って、普段の 7 割のアクセスで区別する、とするならば、5kB をカウントのためのサイズ閾値と決定する、といったことが考えられる。

続いて、サイズ閾値を決定した後、パラメータ閾値を設定する必要がある。この閾値は、これを越えた参照率である場合のほとんどの場合において、サイズを考慮しないアルゴリズムの方が、バイトヒット率が高くなるように決定されなければ、キャッシュの効果を引き出すことができない。この決定に対しては、運用される環境に依存した利用度や分布にも影響を受け、決定的な値を選定することは非常に困難な問題であると思われる。キャッシュサーバ、例えば squid では、

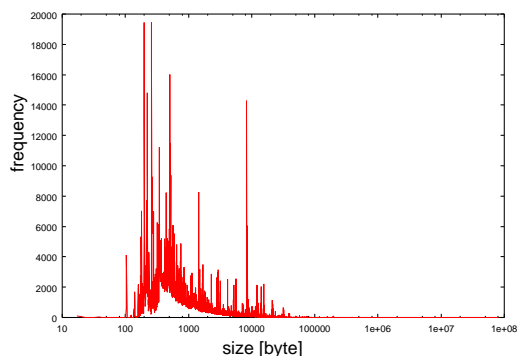


図 4 サイズ別参照 / 頻度分布 東大 ECC サイト、2001 年 1 月 15 日から 18 日

数多くのパラメータ設定を環境に応じて変更できるようになっており、これらのパラメータの設定値によりキャッシュの性能は異なってくるため、サイト状況に適したフィードバックが必要と考えられる。

4. 動的選択手法の評価

幾つかのアクセスパターンを作成し、動的選択手法の評価を行った。前述の LRU、LFU、SF、そして動的選択 (図においては switch として示す) による結果を比較した。

4.1 人工アクセスモデルを用いたシミュレーション評価

本研究においては、幾つかの疑似 workload を用いたシミュレーション評価を行った。評価に用いる workload に関しては、実ログのトレース、ベンチマークツールによる評価等が考えられる。実際に、実ログのトレースをおこなうことは、そのログが収集されたサイトの改善に対して最も有効な手段であり、ユーザ数、アクセス数が多いほど、一般的な特徴を示していると考えられる。しかし、実ログの利用は、将来予測に対して有効であるかどうかは疑わしく、また環境に依存したものである、といった問題が残る。逆にベンチマークツールでは、特殊な環境の想定や、想定パターンのモデル作成により、実際に得られるパターンと異なるパターンに対しても評価が可能となる。しかし、WWW の利用傾向のような場合、現実のような複雑なアクセスパターンを発生することは非常に困難であり、またできるだけ公正な評価を行う上で、注意を要する。

4.2 アクセスモデル

大サイズオブジェクト取得率が比較的変動するモデルに対する評価を示す。アクセスモデルの大サイズオブジェクト、即ち 500KB 以上のサイズのオブジェクト



図 5 500k バイト以上のオブジェクトの参照率の時間推移

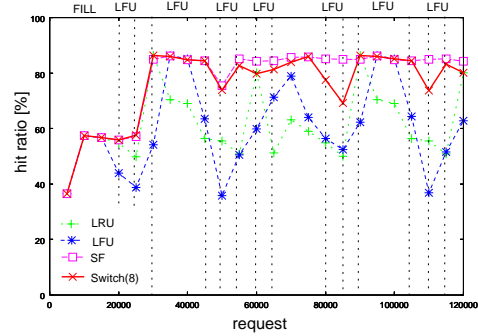


図 8 5000 リクエスト毎のヒット率。図中の LFU, SF は、選択手法により選択されたアルゴリズム、switch(8) は 500k バイト以上コンテンツへの参照率 8%を閾値とした切替えを示す

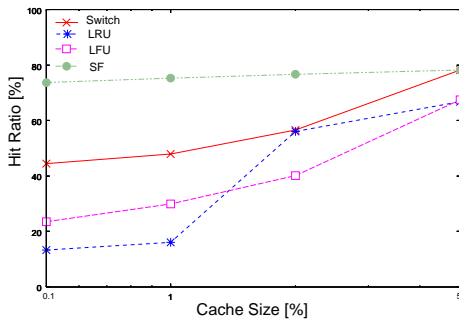


図 6 キャッシュサイズに対する累積ヒット率。キャッシュサイズを全トラフィックに対する相対サイズで示す

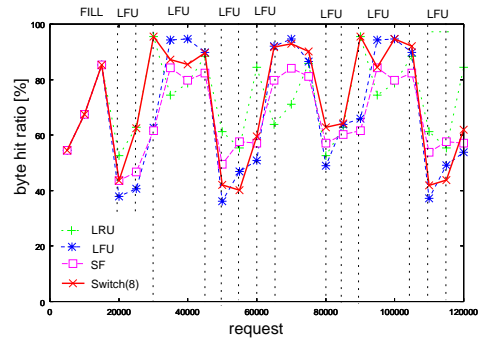


図 9 5000 リクエスト毎のバイトヒット率。図中の LFU, SF は、選択手法により選択されたアルゴリズム、switch(8) は 500k バイト以上コンテンツへの参照率 8%を閾値とした切替えを示す

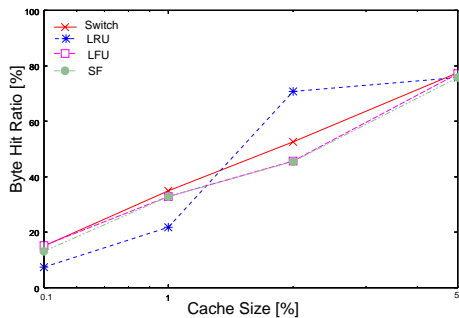


図 7 キャッシュサイズに対する累積バイトヒット率。キャッシュサイズを全トラフィックに対する相対サイズで示す

への参照率の時間的変化が図 5 である。総トラフィックは約 1.7GB であった。

4.2.1 結果

前述のアクセスモデルに対し、キャッシュサイズを相対的に変化させて、ヒット率、バイトヒット率を調査したものが、図 6、図 7 である。切替えの閾値を、500k バイト以上のサイズコンテンツへの参照率 8%としている。ヒット率においては、大サイズオブジェクトも常にキャッシュするアルゴリズムよりも高い値を示した。バイトヒット率においても各アルゴリズム

を単一で用いたときよりも高い値を示した。また、図 8、図 9 に、各 5000 リクエスト毎のヒット率、バイトヒット率の様子を示す。ヒット率においてはサイズを考慮したアルゴリズム (SF) に迫る値を示し、バイトヒット率では、大サイズオブジェクト参照が多い期間において、高い値を示している。このことから、図 5 に示されるような短期的変動に比較的よく追従しているといえる。

本評価においては、利用状況に応じて大サイズオブジェクトが多く要求され、ネットワーク帯域に影響が大きいときには大サイズオブジェクトをキャッシュしその再利用に備えること、逆にそのようなコンテンツの利用頻度が下がれば、小さくて利用頻度が高いものためのスペースをつくり、ヒット率を上げることでキャッシュの効果を、高める、といった、期間毎のアクセス状況に対応することで、ヒット率、バイトヒット率を共に高めるキャッシングの実現がみられた。図 10 に一例として、前述のモデルに対しヒット率、バイトヒット率を共に考慮した場合の評価軸での、比較

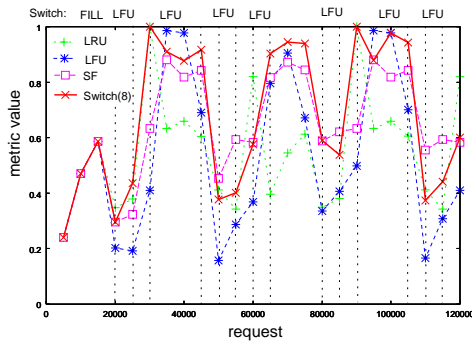


図 10 期間内ヒット率、バイトヒット率を共に考慮した評価 switch(8) は 500k バイト以上コンテンツへの参照率 8%を閾値とした切替えを示す

を示す。ただし、期間内ヒット率を HR_{term} 、その、全アルゴリズムの全期間中の最大値を $HR_{term-max}$ 、バイトヒット率 BHR に関しても同様にし、次式で表される相対的な評価軸を用いた。

$$metric_{term} = \frac{HR_{term}}{HR_{term-max}} \frac{BHR_{term-max}}{BHR_{term-max}} \quad (2)$$

ただし、

$$HR_{term-max} BHR_{term-max} = 1 \quad (3)$$

5. 結 論

本研究では、既存の技術であるキャッシング技術をもとに、多種多様なコンテンツが WWW サービスを通してやりとりされるときのネットワーク利用を軽減することを考えた。現在の置換アルゴリズム決定は通常は統計情報に基づいており、アクセス状況にあまり柔軟ではないといえるため、本研究においては複数アルゴリズムをアクセス状況に従って予測、選択し、リアルタイムのアクセス状況に敏感なキャッシング方法を提案し、検討をおこなった。この提案手法に対し、様々な、単純化したパターンのアクセスモデルを想定、作成し、シミュレーションを行うことで、単一の代表的なアルゴリズムとの比較、評価を行った。現在の多くのアルゴリズムは、相反するヒット率、バイトヒット率の向上に関し、どちらか一方を最適化する目的のものが多いが、アクセス傾向に応じた切替えにより、ヒット率、バイトヒット率、をとともに考慮した場合、その適したアクセスパターンが見られる期間において、上昇させることができる可能性を示した。

5.1 今後の課題

キャッシュ効率に影響力のあるパラメータの組合せにより、評価値は際限無く変動することが予想されるが、細かいパラメータ値の変化、アクセスパターンの変化に対し、より有効な値を探求することは重要で

あると思われる。ただしこれらの設定は、アクセスパターン、環境的影響の複雑さ等のため、最適解を求めることは非常に困難である。また、本質的な問題として、コンテンツ分布や、それに伴うオブジェクトサイズの分布状況、更には環境要素の影響などが、どのように変化するのか、または変化しないのか、ということによっても、この手法が有効であるのかどうかということも変わってくる。従って、特にサイズ分布の様子を考慮にいれた検討が重要となるとと思われる。

参 考 文 献

- 1) <http://www.akamai.com/>.
- 2) <http://www.inktomi.com/>.
- 3) <http://www.cacheflow.com/>.
- 4) Junichi Funasaka, Keizo Saisho, Akira Fukuda, "Automatic Selecting of Required NetNews Articles, The 6th Asia-Pacific Software Engineering Conference (APSEC'99), pp.88-91, 1999.
- 5) Arun Iyenger, Jim Challenger, Daniel Dias, and Paul Dantzig, "High-Performance Web Site Design Techniques", IEEE Internet Computing, Volume:4 Issue:2, March-April 2000.
- 6) 総務省 「情報通信白書平成 13 年度版」 <http://www.soumu.go.jp/hakusyo/tsushin/index.html>
- 7) Ari Luotonen and Kevin Altis, "World-Wide-Web Proxies", Proceedings of First International World-Wide Web Conference, April 1994.
- 8) "Squid Web Proxy Cache", <http://www.squid-cache.org/>
- 9) "Toplevel Cache Server Survey",
- 10) Anawat Chankhunthod, et al, "A Hierarchical Internet
- 11) Stephen Williams, Marc Abrams, Charles R. Standridge, et al., "Removal Policies in Network Caches for World-Wide Web Documents", Proceedings of the ACM SIGCOMM'96 conference.
- 12) M. Arlitt et al., "Evaluating Content Management Techniques for Web Proxy Caches", Proc. Second Workshop Internet Server Performance Evaluation Review, Aug. 1999.
- 13) J Pei Cao and Sandi Irani, "Cost-aware WWW proxy caching algorithms", in Proceedings of the 1997 USENIX Symposium on Internet Technology and Systems, December, 1997, pp.193-206.