

修 士 論 文

題 目

プロキシサーバによる効率的な
Web閲覧履歴の取得に関する研究

指導教員

報 告 者

栗 原 聖 治

岡山大学 大学院自然科学研究科 電子情報システム工学専攻

平成 22 年 2 月 10 日 提出

要約

利用者が計算機を用いて作業を行う場合，そこには利用者固有の利用傾向が存在する．計算機が利用者の利用傾向を把握することで，利用者にとってより利便性の高いサービスを提供できる．利用傾向を把握する方法として，例えば，利用者のキー入力履歴やプログラムの起動履歴を利用することが考えられる．また，ネットワークの普及により，ネットワークを介した通信を利用する作業が増えてきているため，通信の履歴も利用者の利用傾向を把握するための有用な情報となる．

そこで，Web 閲覧履歴を把握するために，Private Proxy Server という，利用者計算機上で動作する個人専用のプロキシサーバが提案された．Private Proxy Server は外部計算機との通信を仲介し，通信履歴を保存する．Private Proxy Server を用いることにより，利用者計算機上の情報を外部に漏らすことなく，計算機外部との通信履歴を扱える．しかし，Private Proxy Server により保存された履歴情報から利用者の Web 閲覧履歴を把握するためには，利用者毎に異なった利用傾向の把握や特殊なアクセスパターンに対応する必要がある．本論文では，これらの問題に対処した新しいアクセスパターン解析を提案した．具体的には，参照履歴のツリー化によって特殊なアクセスパターンに対処し，フィルタへのパーソナライズ機能の追加を提案した．これによって，特殊なアクセスパターンにおいて再現率が 64 % から 91 % に向上したことを示した．

目次

1	はじめに	1
2	プロキシサーバによる Web 閲覧履歴の取得	2
2.1	Private Proxy Server の通信履歴	3
2.2	目的	3
2.3	提案手法	4
3	通信履歴と Web ページ構成	6
3.1	通信履歴	6
3.2	Web ページの構成	6
3.2.1	一般的な Web ページ	6
3.2.2	単一 URL のみの Web ページ	8
3.2.3	深い階層構造を持つ Web ページ	8
3.3	通信履歴と Web ページ構成のまとめ	8
4	Web ページの構造に着目したフィルタ	10
4.1	抽出する閲覧情報	10
4.2	注目するパラメータ	10
4.3	フィルタリング条件	13
4.4	閾値の設定方針	13
4.4.1	設定基準	14
4.4.2	閾値の評価	15
4.5	実験	15
4.5.1	実験用データ	16
4.5.2	実験結果	16
4.6	課題	16

4.7 対処	17
5 Web ブラウジングの流れを扱うフィルタ	18
5.1 ブラウジングツリーの構成	18
5.2 ブラウジングツリーの閲覧時間推測	20
5.3 ブラウジングツリーの実装と評価	21
5.3.1 評価方式	21
5.3.2 評価結果	21
5.3.3 考察	21
6 おわりに	24
謝辞	25
参考文献	26

図 目 次

2.1	Private Proxy Server のシステム構成	3
2.2	Private Proxy Server の通信履歴の構成	4
2.3	Private Proxy Server の通信履歴	4
3.1	一般的な Web ページ	7
3.2	URL1 の Web ページを閲覧後, 文章中の URL4 を click した例	7
3.3	深い階層構造を持つ Web ページの URL 参照関係	9
3.4	誤った URL 参照関係	9
4.1	履歴の時間差に注目した図	12
4.2	再現率と適合率	14
4.3	機械参照閾値の再現率と適合率	14
4.4	閲覧閾値の再現率と適合率	15
5.1	基本構成	19
5.2	Node の追加ルール	19
5.3	閲覧時間の推測方式	20
5.4	抽出データの分布	22

第 1 章

はじめに

利用者が計算機を用いて作業を行う場合，そこには利用者固有の利用傾向が存在する．計算機が利用者の利用傾向を把握することで，利用者にとってより利便性の高いサービスを提供できる．利用傾向を把握する方法として，例えば，利用者のキー入力履歴やプログラムの起動履歴を利用することが考えられる．また，ネットワークの普及により，ネットワークを介した通信を利用する作業が増えてきているため，通信の履歴も利用者の利用傾向を把握するための有用な情報となる．

そこで，Web 閲覧履歴を把握するために，Private Proxy Server という，利用者計算機上で動作する個人専用のプロキシサーバが提案された [1]．Private Proxy Server は外部計算機との通信を仲介し，通信履歴を保存する．Private Proxy Server を用いることにより，利用者計算機上の情報を外部に漏らすことなく，計算機外部との通信履歴を扱える．しかし，Private Proxy Server により保存された履歴情報から利用者の Web 閲覧履歴を把握するためには，利用者毎に異なった利用傾向の把握や特殊なアクセスパターンに対応する必要がある．本論文では，これらの問題に対処した新しいアクセスパターン解析を提案した．具体的には，参照履歴のツリー化によって特殊なアクセスパターンに対処し，フィルタへのパーソナライズ機能の追加を提案した．これによって，特殊なアクセスパターンにおいて再現率が 64 % から 91 % に向上したことを示した．

本論文の以降では，第 2 章にてプロキシサーバの生成する通信履歴とは何かについて述べ，本研究の目的について述べる．次に第 3 章では通信履歴と Web ページ構成について述べる．次に，第 4 章では Web ページの構造に着目して作成したフィルタについて述べ，その結果判明した課題について述べる．そして，第 5 章において，課題への対処として Web ブラウジングの流れを扱う方式を述べ，実装と評価を述べる．第 6 章では本論文のまとめと残った課題について述べる．

第 2 章

プロキシサーバによる Web 閲覧履歴の取得

本章では、通信履歴保存手段として、プロキシサーバである Private Proxy Server について述べ、Private Proxy Server の計算機利用履歴を利用する上での課題と対処について述べる。

Private Proxy Server とは、利用者計算機上で動作する Apache HTTP Server の個人用プロキシサーバである。Private Proxy Server は利用者支援を目的とし、利用者による計算機利用や通信を監視し、計算機利用履歴を保存、その情報を利用し、利用者にとって利便性の高いサービスを提供することを目的としている。

図 2.1 に Private Proxy Server の利用形態を示し、以下に説明する。Private Proxy Server は、AP と計算機外部との通信を仲介し通信履歴を保存し、データベース (以下 DB) に保存する。そして、Private Proxy Server の計算機利用履歴を利用する AP は DB を参照し、利用者にサービスを提供する。ここで、Private Proxy Server の計算機利用履歴を利用する AP を適応制御 AP と呼び、それ以外の AP を一般 AP と呼ぶ。

Private Proxy Server の利点は、利用者計算機上で動作するため、情報を外部に漏らさないことや、プロキシサーバとして動作するので、導入にあたって特別な環境を用意する必要がないことがあげられる。

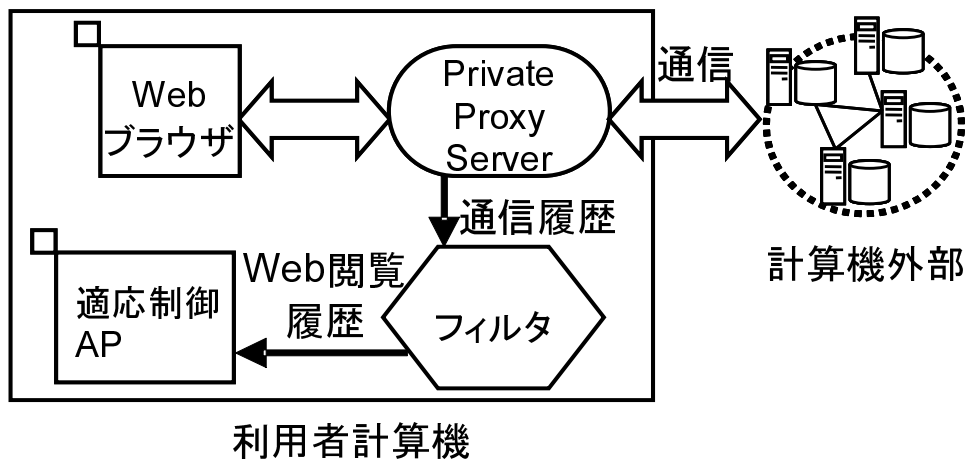


図 2.1 Private Proxy Server のシステム構成

2.1 Private Proxy Server の通信履歴

Private Proxy Server の保存する通信履歴とは、利用者が Web サーバへアクセスしたときに得られる情報を保存したものである。ここでは Private Proxy Server の保存する通信履歴を図 2.2 に示し説明する。

Private Proxy Server の通信履歴は Apache HTTP Server の Common Log Format[2] に則っており、その内容は、(1) 参照した日時、(2) Content-Type、(3) サービス状態コード、(4) 受信したバイト数、(5) リファラ、(6) 参照した URL(以下リクエスト) である。

通常、利用者が Web ページを閲覧した場合、Web ページ内には画像や HTTP リダイレクト、フレームとして表示すべき Web 断片への参照といった要素が含まれている。そのため、1 つの Web ページを閲覧した場合でも、複次的に多数の URL を参照していることとなる。したがって、1 つの Web ページに対する通信履歴には複数のエントリが記録されることがある。

2.2 目的

本研究の目的は、通信履歴から利用者の Web 閲覧履歴を取得することである。そのためには、利用者が実際に Web ブラウザで閲覧したページの URL 情報を得る必要がある。ここで、利用者が実際に Web ブラウザで閲覧したページの URL 情報とは以下の条件を満たすものとする。

[06/Mar/2009:15:32:43 東京 (標準時)] "text/html" 200 9608
(1) (2) (3) (4)

http://www.example.com/
(5)

-> http://www.it.example.com/index.html
(6)

図 2.2 Private Proxy Server の通信履歴の構成

- (1) localhost - - [13/Nov/2006:11:32:38 東京 (標準時)]
"GET http://www.it.example.com/index.html HTTP/1.1" 200 10790
http://www.example.com/
-> http://www.it.example.com/index.html
- (2) localhost - - [13/Nov/2006:11:32:38 東京 (標準時)]
"GET http://www.it.example.com/title.jpg HTTP/1.1" 200 22703
http://www.it.example.com/index.html
-> http://www.it.example.com/title.jpg

図 2.3 Private Proxy Server の通信履歴

条件 1 Web ページを構成するトップの URL である

条件 2 利用者が時間をかけて閲覧した URL である

例として、図 2.3 の通信履歴から、利用者が閲覧した Web ページを把握するとすると、利用者が実際に参照した URL とは、(1) の Web ページ閲覧時に参照された URL である。また、(2) の履歴は (1) の履歴の時に参照した Web ページに付随する URL を参照した時の履歴なので、不要な情報となる。

このように、Private Proxy Server の保存する通信履歴から、必要な情報を抽出する方法が必要となる。

2.3 提案手法

そこで、本研究では Private Proxy Server の計算機利用履歴から必要な情報を抽出するフィルタを作成することで目的達成することを提案する。フィルタを作成する方法であれば、

Private Proxy Server や適応制御 AP に手を加えることなく情報の抽出を行うことができる。

以降では、作成するフィルタの例として、図 2.3 で示した Private Proxy Server の通信履歴から、利用者が実際に Web ブラウザで閲覧したページの URL 情報を抽出するフィルタの開発を検討する。

第 3 章

通信履歴と Web ページ構成

本章では，Private Proxy Server の通信履歴と Web ページの構成の関係について述べる．

3.1 通信履歴

2.1 節にて述べたとおり，1 つの Web ページを閲覧した場合でも，複次的に多数の URL を参照していることになり，通信履歴には複数のエントリが記録されることがある．

そこで，Web ページの構成と，その Web ページを閲覧することによって記録される通信履歴との関係を把握することが必要となる．そうすることで実際に利用者が閲覧した Web ページのみを抽出するフィルタリング方式を検討することができる．

3.2 Web ページの構成

ここでは，Web ページの構成と通信履歴の関係について述べる．

3.2.1 一般的な Web ページ

一般的な Web ページというのは図 3.1 のように，画像など他のファイルを参照している形式を取っている．こういう Web ページを参照すると，それぞれの URL の参照関係は図 3.2 のように示される．

このとき URL1 ~ URL3 の参照順は，通信履歴の日時から把握でき，URL の参照状態はサービス状態コードから把握できる，また通信履歴のリクエストが URL2 の時，そのリファ

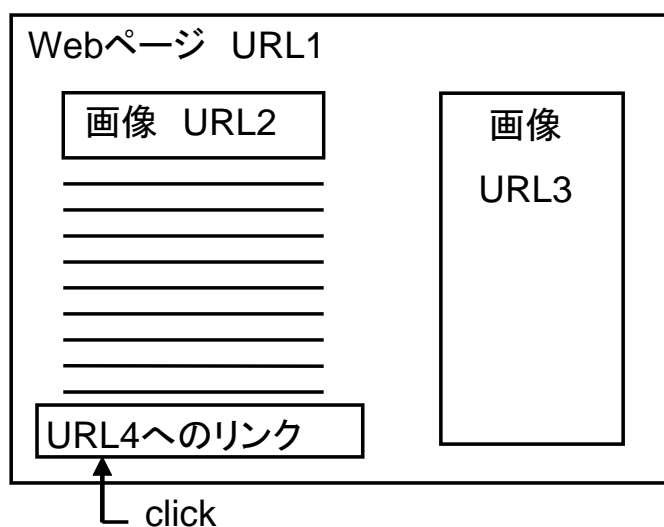


図 3.1 一般的な Web ページ

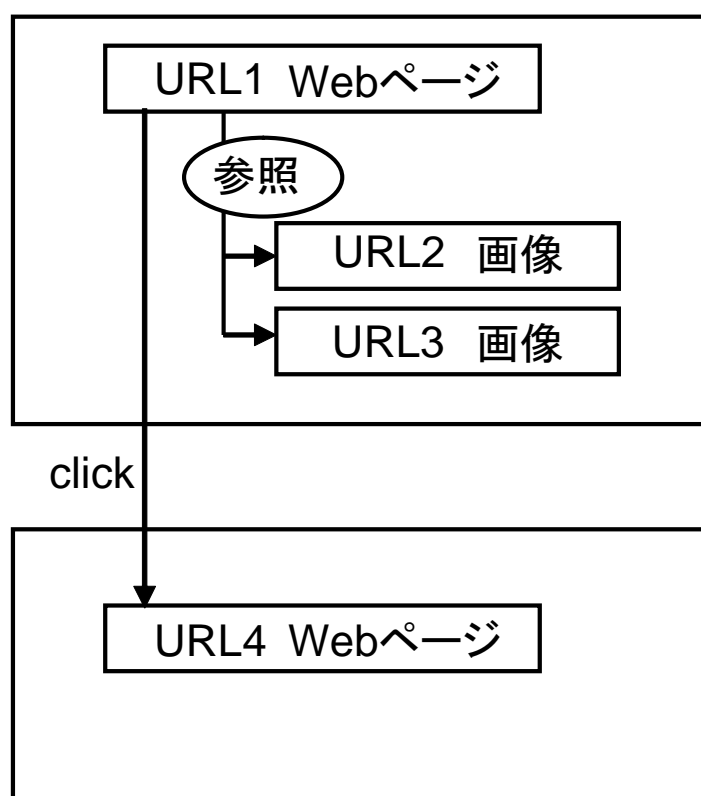


図 3.2 URL1 の Web ページを閲覧後，文章中の URL4 を click した例

ラは URL1 であり，リクエストが URL3 の時も同様に，URL1 がリファラである．このことからリファラは Web ページにおける階層構造を示す情報として扱うことができる．しかし，URL1 にて別の Web ページの URL4 をリクエストとした場合のリファラは URL1 となるので，リファラが一概に Web ページの階層構造を表しているわけではない．

以後，Web ページの階層構造のトップにある URL(図 3.2 における URL1) を MainURL と呼び，それ以外の URL(図 3.2 における URL2，URL3) を SubURL と呼ぶこととする．

利用者が実際に Web ブラウザで閲覧したページの URL 情報はこの MainURL である．よって MainURL と SubURL を判別する必要がある．

3.2.2 単一 URL のみの Web ページ

単一 URL のみの Web ページとは，例えばテキストのみの Web ページのように，MainURL で示されるファイルのみで構成された Web ページのことである．

単一 URL のみの Web ページの場合，MainURL から別の Web ページへ移動しない限り，MainURL はリファラになることはなく，通信履歴では参照されたときのリクエストとしてのみ保存される．

3.2.3 深い階層構造を持つ Web ページ

深い階層構造を持つ Web ページとは，図 3.3 が示すように SubURL が SubURL をリクエストする構成の Web ページである．このような構成は閲覧履歴に記されたリクエストの関係からは図 3.4 のような構成にも見える為，リファラのみで階層構造を完全に判別することはできない．

3.3 通信履歴と Web ページ構成のまとめ

3.2 節で述べたように，Web ページを構成する URL は階層構造をもっており，その階層構造が通信履歴のリファラから伺い知ることができる．階層構造の TOP である MainURL はこのリファラとして通信履歴に記録される可能性が高いと考えられる．

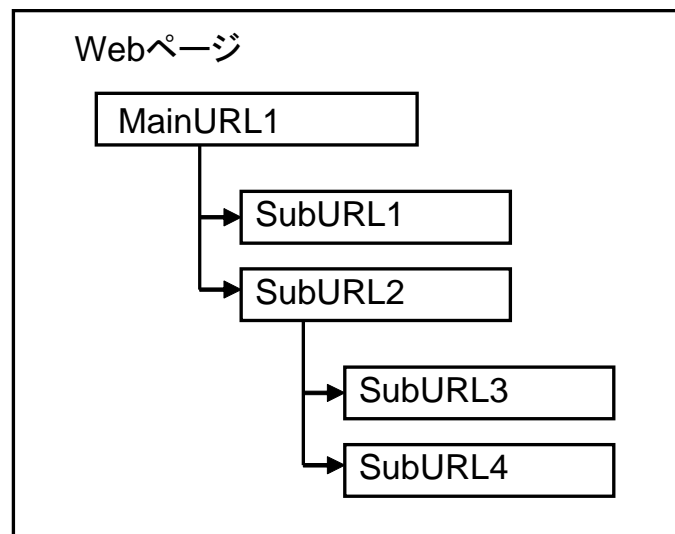


図 3.3 深い階層構造を持つ Web ページの URL 参照関係

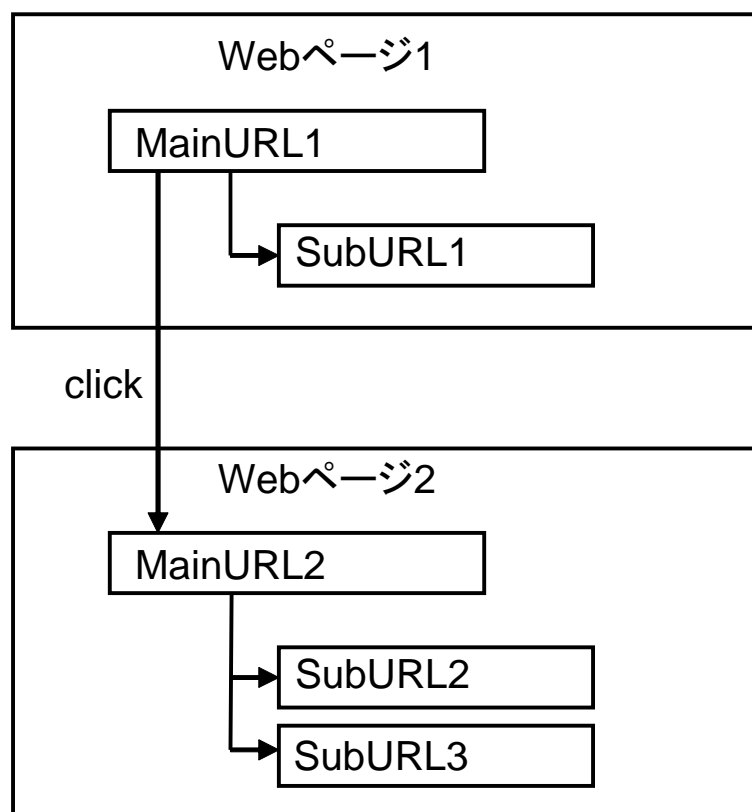


図 3.4 誤った URL 参照関係

第 4 章

Web ページの構造に着目したフィルタ

本章では，Web ページの構造に着目したフィルタについて述べる．

4.1 抽出する閲覧情報

ここでは，Private Proxy Server が保存した Web 通信履歴から，どのような考えに基づいて履歴情報を抽出するか述べる．

Web 通信履歴の内，必要とされる情報は，利用者が直接参照した Web ページの URL の参照履歴である．つまり，Web ページの階層構造の TOP である MainURL が必要とされる情報である．

また，利用者が閲覧した Web ページの URL を保存したいので，閲覧時間の短い Web ページの URL は，利用者にとって重要な履歴ではないと考え，保存しない．

4.2 注目するパラメータ

ここでは，閲覧情報の抽出に関して注目するパラメータについて述べる．

リファラ 3.3 節で述べたように，リファラは Web の階層構造を表すので，リファラは Main-URL を判別する上で，強力な要素だと考えられる．そこで Private Proxy Server の通信履歴のリファラに注目し，リファラとなった URL を利用者が閲覧した URL として抽出する方式でフィルタを開発し，通信履歴をフィルタリングした．しかし，リファラのみで MainURL を判別すると，3.2.2 項で述べたように，単一 URL のみの Web ページ

がリファラとして記録されておらず，MainURL と判別されないという問題と，図 3.3 のような深い階層構造を持つ Web ページを，図 3.4 のような構造の複数の Web ページだと誤認するという問題が生じる．

また，Private Proxy Server の通信履歴においてリファラが”-”となっている履歴があるが，これは利用者がブックマークや URL の直接入力によって URL を参照したことを表している．このとき参照された URL は利用者が閲覧した URL と考えられる．

日時 URL を参照した時の時間差は，利用者がその URL を閲覧してした時間と考えられる．履歴の時間差に注目した図を図 4.1 に示す．それぞれの四角の右下の数値はその URL を参照した時間 (秒) である．このとき MainURL 同士の時間差を，利用者が Web ページを閲覧していた時間として考えることができる．図 4.1 において，MainURL1 と MainURL2 の時間差 20 秒を MainURL1 の Web ページの閲覧時間だと考える．この場合，MainURL1 の閲覧時間が充分長いと考えられる．一方，MainURL2 と MainURL3 の時間差 4 秒を MainURL2 の Web ページの閲覧時間だと考えると，MainURL2 の参照時間は短いと考え，MainURL2 の Web ページは利用者にとって重要ではないのではないかと考える．連続した MainURL 同士の時間差を調べるためには MainURL と SubURL を判別する必要がある．また，Web ブラウザが MainURL から SubURL を参照する速度は，利用者が直接 MainURL を参照する速度より充分に早いと考えられる．このことは，3.2.3 項に示した深い階層構造を持つ Web ページの MainURL を判別する要因となる．

サービス状態コード サービス状態コードは HTTP/1.1[3] で規定されており，ここではユーザが閲覧した Web ページを抽出したいので，サービス状態コードが”OK”を表す”200”のときと，”Not Modified”を表す”304”の通信履歴のみを扱う．

Content-Type Content-Type は取得したデータの形式を表している．ここで，MainURL と考えられる履歴の Content-Type は”text/html”であるので，Content-Type が”text/html”の履歴を扱う．

これにより，本来 SubURL であるが，さらに SubURL を参照するため，リファラとして記録されるデータ (例えば CSS や XML) を MainURL だと誤認しなくなる．

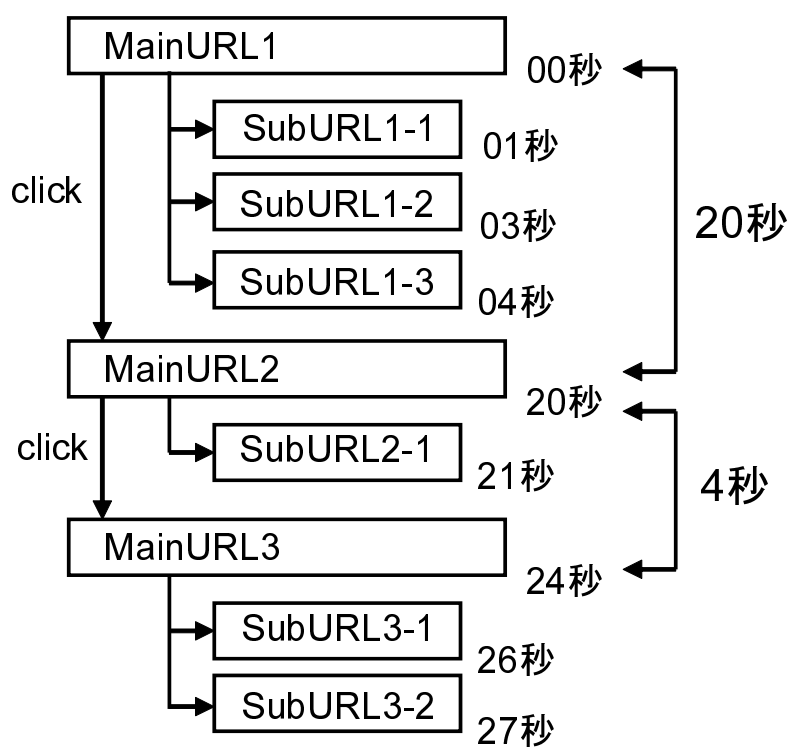


図 4.1 履歴の時間差に注目した図

4.3 フィルタリング条件

4.2 節より，日時・リファラ・サービス状態コード・Content-Type を抽出する Web 閲覧情報を判別するパラメータとしてフィルタを作成する．抽出法のフローチャートを以下に示す．

- (1) リファラとなっている URL を抽出し，そのリファラがリクエストされた時の履歴を MainURL 参照履歴候補とする．リファラが重複している場合，先に参照された履歴のみを扱う．またリファラが“-”である履歴は後で扱うのでここでは除外しておく．
- (2) 閲覧 URL 履歴候補からサービス状態コードが“200”もしくは“304”以外の履歴を除外する．
- (3) 残った閲覧 URL 履歴候補から Content-Type が“text/html”ではない履歴を除外する．
- (4) 深い階層構造を判別するために，MainURL 参照履歴候補とその 1 つ前の履歴との参照時間差を見て，両者を参照した時の時間差が利用者が URL を直接参照する間隔より十分に短い場合，その MainURL 参照履歴候補は深い改造構造の TOP ではないと判別し，MainURL 参照履歴候補から除外する．
- (5) MainURL 参照履歴候補にリファラが“-”の時の履歴を加え，MainURL 参照履歴候補同士の参照時間差を利用者の閲覧時間と見て，利用者の閲覧時間が十分に長ければ，その MainURL 参照履歴候補を閲覧 URL 履歴候補と判定する．このとき，一番最後の MainURL 参照履歴候補は利用者が最後に閲覧した URL なので閲覧 URL 履歴候補と考える．

このとき，利用者が URL を直接参照する間隔の閾値や利用者が経由ページを閲覧した時間の判定に利用する閾値を設定する必要がある．閾値の設定については，4.4 節で述べる．

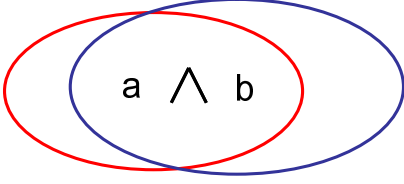
4.4 閾値の設定方針

本節では，閾値の設定基準について 4.4.1 項で述べ，実際の閾値の決定を 4.4.2 項で述べる．

また以降では，利用者が URL を直接参照する間隔の閾値を機械参照閾値と呼び，利用者が経由ページを閲覧した時間の判定に利用する閾値を閲覧閾値と呼ぶ．

$$\text{データの再現率} = \frac{a \wedge b}{a}$$

$$\text{データの適合率} = \frac{a \wedge b}{b}$$



 (a) = 正解データ
 (b) = 抽出したデータ

図 4.2 再現率と適合率

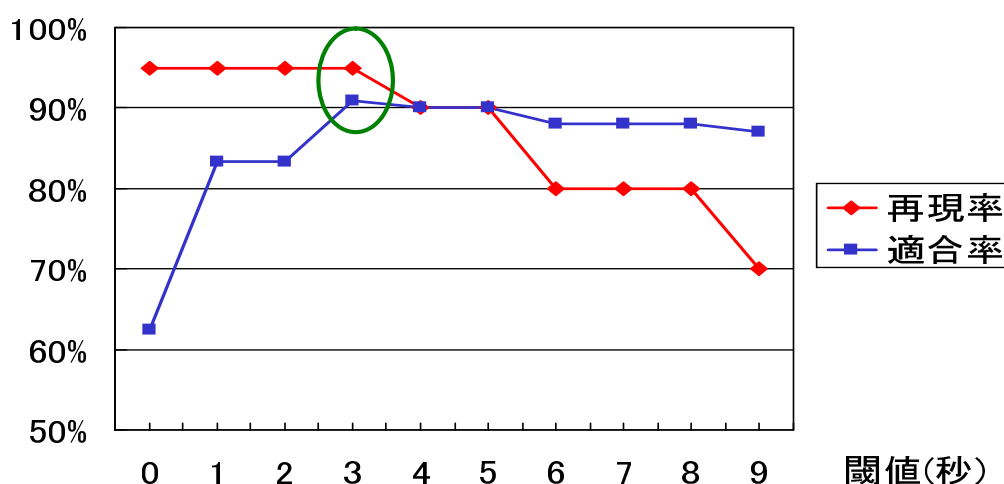


図 4.3 機械参照閾値の再現率と適合率

4.4.1 設定基準

閾値の設定において、フィルタリング後のデータの再現率と適合率に着目する。ここで、再現率とは、抽出したデータの正解データの抽出漏れの少なさを表し、適合率とは、抽出したデータにおける、正解データの割合、つまり誤ったデータの少なさを表すものとする。

機械参照閾値における正解データとは MainURL のことであり、閲覧閾値における正解データとは利用者が時間をかけて閲覧した MainURL のことである。ユーザの Web 閲覧履歴を把握する上では、正解データの漏れの少なさが優先されると考えられるので、再現率を優先し、再現率を保ちながら、適合率を向上できる閾値が望ましいと考える。

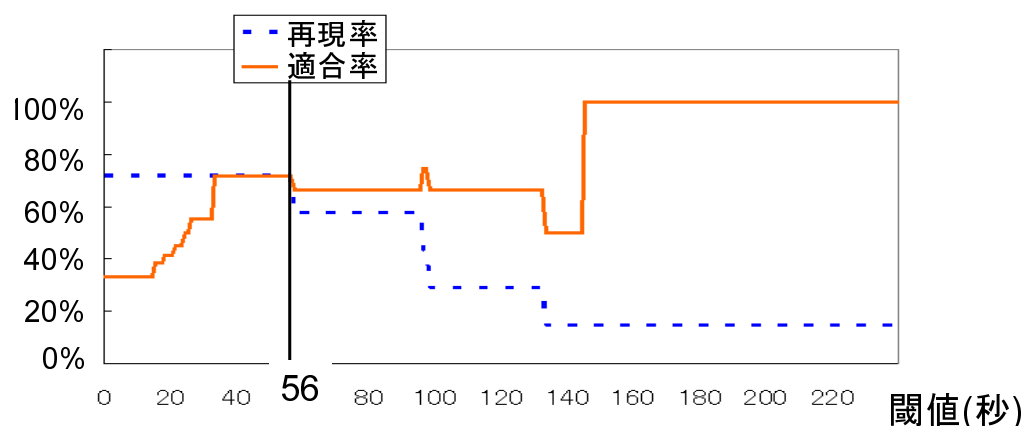


図 4.4 閲覧閾値の再現率と適合率

4.4.2 閾値の評価

閾値を 1 秒単位で変えながら、30 分間ほど調べものを行った際の通信履歴をフィルタリングし、機械参照閾値の再現率と適合率を測定した結果を図 4.3 に、閲覧閾値の再現率と適合率を測定した結果を図 4.4 が示す。

これらの図から、再現率と適合率は機械参照閾値の場合も閲覧閾値の場合でもトレードオフの関係であることが分かる。4.4.1 項で述べたように、再現率を保ちながら、適合率を向上できる閾値が望ましいので、機械参照閾値を 3 秒に、閲覧閾値を 56 秒が妥当だと考えられる。

4.5 実験

4.4.2 項から、機械参照閾値を短くしても MainURL の取り逃しがあった。取り逃しのあった部分の履歴はタブブラウザによって、まず複数の閲覧候補を開いてから閲覧を行った場合であった。

そこで、この特殊なアクセスパターンで Web ブラウジングを行ったデータに着目して実験を行った。

4.5.1 実験用データ

使用する実験用のデータは，タブブラウザを利用した閲覧形態で Web ブラウジングを行ったもので，全データ数は 248 個，そのうち本当に閲覧したデータは 11 個である．また，4.4.2 項から，機械参照閾値を 3 秒に，閲覧閾値を 56 秒に設定した．

4.5.2 実験結果

評価結果を表 4.1 に示す．

フィルタ	抽出データ数	抽出正解データ数	再現率	適合率
Web ページの構造に着目したフィルタ	8 個	7 個	64 %	88 %

表 4.1 実験結果

4.6 課題

Web ページの構造に着目したフィルタでは，特殊なアクセスパターンにおいて，再現率が低くなってしまった．また，利用者毎に異なった利用傾向を把握する上で，以下の課題を発見した．

(1) タブブラウザの利用を考えていない

現在，多くの Web ブラウザはタブブラウザとなっている．タブブラウザを利用して作業する場合，利用者はまず検索ページで特定のキーワードについて検索し，検索結果から目的に合致していそうなページを予めタブに開いてから，ページを閲覧するといった利用方法がとられる．このような利用方法を考えていないため，正解データの見逃しが多く発生する．

具体的には，既存フィルタは，MainURL 参照履歴候補とその 1 つ前の履歴との参照時間差を見て，両者を参照した時の時間差が利用者が URL を直接参照する間隔より充分に短い場合，その MainURL 参照履歴候補は深い改造構造の TOP ではないと判別している．しかし，タブブラウザを利用した形態の場合，MainURL を短い間隔で参照していくことが多く発生するため，正解データの見逃しが多く発生する．

(2) 閾値や保存するデータは利用者の利用傾向に大きく左右される

閾値は利用者の利用傾向によって変わり、保存したいデータの種類、つまり Content-Type も利用者によっては PDF の閲覧履歴を保存したいという要求を持つ利用者もいれば、PDF の閲覧履歴を保存したくはないという利用者もいる。そのため、一意に閾値や、保存対象となるデータの種類を決定できない。

(3) フィルタリング後のデータに利用者が手を加えられない

システムが完全にフィルタを行えるわけではない。また、フィルタリング自体は完全な場合でも、利用者は保存するつもりのない情報（例えば、休憩時間に行った Web 閲覧履歴）が含まれる場合もある。しかし、フィルタリング後のデータを利用者が確認し、操作する方法がない。

(4) データ収集期間中の Web ブラウジングを一面的にしか扱えない

1 回のデータ収集中に異なる 2 つ以上の仕事を行った際に、それらを判別することが出来ない。

4.7 対処

対処 1 利用者の Web ブラウジングを推測するために、Web ブラウジングの流れ扱う

Web ブラウジングの流れを詳細に扱うことで、課題 1 に対処する。対処した結果を 5 章で示す。

対処 2 パーソナライズ機能の追加

閾値や保存する Content-type を利用者が変更できるようにし、課題 2 に対処する。

対処 3 利用者によるデータ操作機能を追加

フィルタリング後のデータを表示し、利用者がデータの扱いを操作できるようにし、課題 3 に対処する。

対処 4 Web ブラウジングの流れから仕事を推測

異なる仕事は異なる Web ブラウジングの流れで行われると推測し、課題 4 に対処する。

これらの対処を行うためには、Web ブラウジングの流れを扱うということが不可欠であると考えられる。そこで、以降本論文では、対処 1 の Web ブラウジングの流れを扱うことに関して述べる。

第 5 章

Web ブラウジングの流れを扱うフィルタ

4.6 節の課題から，利用者の Web ブラウジングの流れというものを把握する必要があるのではないかと考えた．そこで本章では，Web ブラウジングの流れを扱うフィルタについて述べる．

5.1 ブラウジングツリーの構成

Web ブラウジングの流れはリファラによる参照関係が木構造をなしていることから，木構造で取り扱うのがよいと考えた．本節では，Web ブラウジングの流れを扱う木構造（以降，ブラウジングツリー）の構成について述べる．通信履歴より，以下のルールでブラウジングツリーを構成する．

- (1) 木の各 Node は通信履歴 1 つに相当する
 - (2) リファラが“-”である通信履歴を木の根とする
 - (3) 木の各 Node はリファラによる親子関係を持つ
 - (4) 複数の木に同じ MainURL があり，その MainURL を親とする Node を追加する場合，最後に更新された木に追加する
- 図 5.2 の例では，Node5 の所属する木のほうが最後に更新された木なので，Node5 の子として追加する．最後に更新された木に追加する理由は，最後に更新された木の流れが，利用者による最新の Web ブラウジングの流れであると考えられるからである．
- (5) 自身の属している木に含まれている URL が，再び参照された場合でも，子 Node として追加する

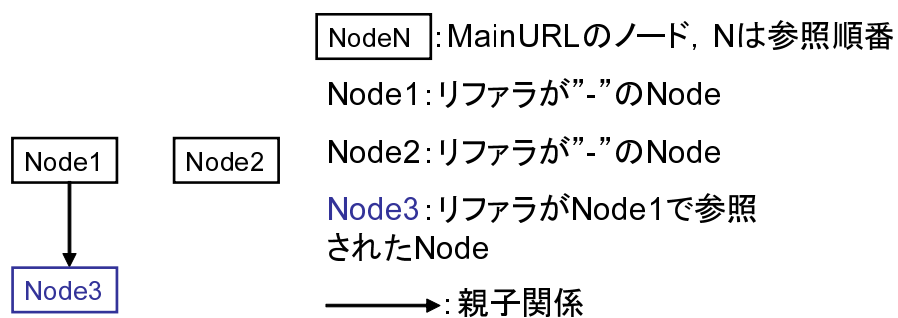


図 5.1 基本構成

(例) Node3とNode5のURLが同じであり, そのURLをリファラとしてもつNode6を追加

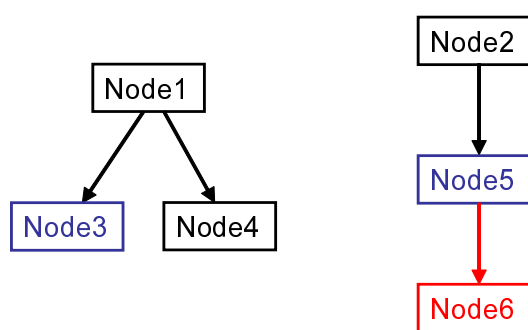


図 5.2 Node の追加ルール

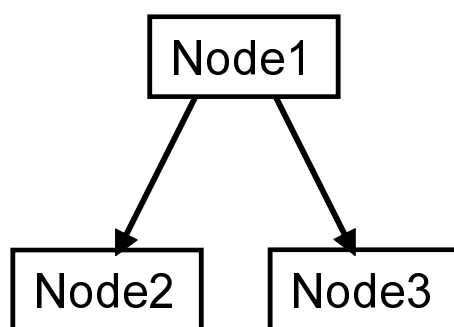


図 5.3 閲覧時間の推測方式

5.2 ブラウジングツリーの閲覧時間推測

まず、自身と自身の親の MainURL の Node との参照時刻差から、深い階層構造を判別する。これにより、Web ページの構造に着目したフィルタの場合と違い、MainURL を短い間隔で参照した通信履歴を、機械参照と誤認するのを防ぐことができる。

MainURL の推測が終わったら、以下の方法で閲覧時間の推測を行う。また、ここで言う Node は MainURL の Node のことを指す。

- (1) 自身の子として追加された Node との参照時刻差を閲覧時間として推測
- (2) 自身の子に Node が存在しない場合は、自身の次に木に追加された Node との参照時刻差を閲覧時間として推測
- (3) もし、自身の親と自身の次に木に追加された Node の親が同じ場合は、タブブラウザを利用した閲覧形態が取られたと推測し、自身の次に木に追加された Node と、その親の Node の参照時刻差を閲覧時間として推測

図 5.3 の場合、Web ページの構造に着目したフィルタでは、Node2 の閲覧時間を Node2 と Node3 の参照時刻の差で推測していたため、タブブラウザを利用した閲覧形態において、Node2 の閲覧時間が短く判別されている。しかし、この方法であれば、Node2 の閲覧時間を既存の方法に比べて長く推測するため、正解データの取り逃しを抑制することが出来る。一方で、図 5.3 の構造はタブブラウザを利用した閲覧形態でなくとも同じ構造をとる場合がある。その場合は、Node2 の閲覧時間を誤って長く推測してしまい、本来は少ししか Node2 を閲覧してなくても正解データと推測することになるが、本研究では再現率がより重要であるため、適合率の低下は大きな問題ではないと考える。

フィルタ	抽出データ数	抽出正解データ数	再現率	適合率
Web ページの構造に着目したフィルタ	8 個	7 個	64 %	88 %
Web ブラウジングの流れを扱うフィルタ	13 個	10 個	91 %	77 %

表 5.1 評価結果

5.3 ブラウジングツリーの実装と評価

5.1 節のブラウジングツリーの構成に基づき，プロキシサーバの通信履歴から，ブラウジングツリーを構成し，5.2 節の閲覧時間推測方法に基づいて利用者の閲覧した URL の推測を行うフィルタを実装した．これにより対処 1 を行った．つまり課題 1 のタブブラウザによる利用に対処したと考えられる．

そこで，4.5 節の実験を新たに実装した Web ブラウジングの流れを扱うフィルタを用いて行い，その結果を評価する．

5.3.1 評価方式

4.5 節で使用した 4.5.1 項の同一の通信履歴を Web ページの構造に着目したフィルタと，新たに実装した Web ブラウジングの流れを扱うフィルタを用いてフィルタリングを行い，4.4.1 項で定義した再現率と適合率で比較した．また，4.4.2 節より，ともに機械参照閾値を 3 秒に閲覧閾値を 56 秒に設定した．

5.3.2 評価結果

評価結果を表 5.1 に示す．

このように再現率の向上が見られる．一方で，5.1 節で述べたとおり，適合率の低下が見られた．

5.3.3 考察

評価結果を図 5.4 を用いて考察する．

B 両方のフィルタで抽出された誤ったデータ

このデータはリダイレクトにより，リファラが”-”で記録されたデータである．通常の HTTP リダイレクトでは，サービス状態コードが 301・302・307 であり除外できるが，

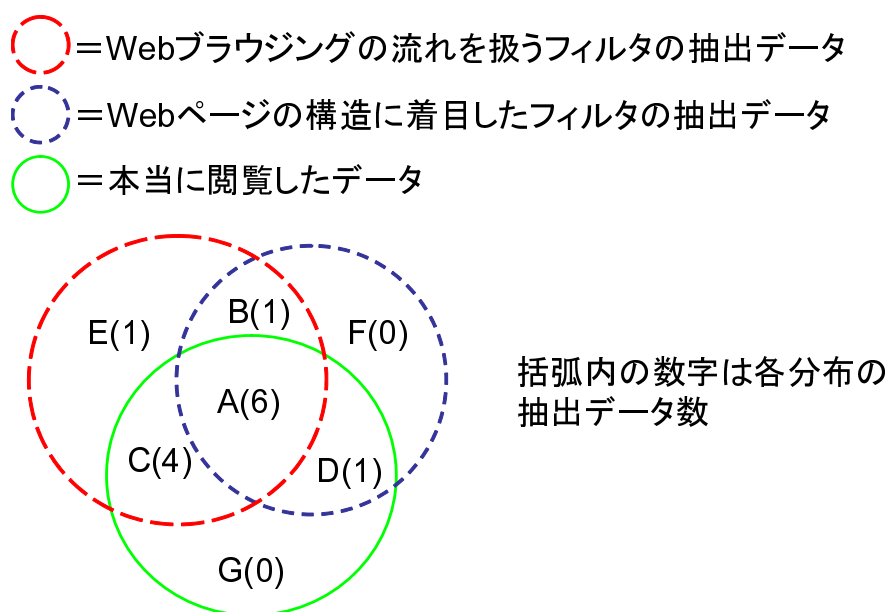


図 5.4 抽出データの分布

META タグやスクリプトでもリダイレクトの記述が出来るため，その場合に抽出されてしまう．

C Web ブラウジングの流れを扱うフィルタでのみ抽出された正解データ

タブブラウザを利用した閲覧形態で記録されるため，Web ページの構造に着目したフィルタでは除外されるデータが抽出できている．

D Web ページの構造に着目したフィルタでのみ抽出された正解データ

これは，タブブラウザを利用した閲覧形態では同時に複数のデータ参照処理が行われるため，参照処理の遅延と複数の MainURL 候補の参照処理が混ざったために記録された正解データだと考えられる．

具体的には，参照処理の遅延のために MainURL1 からの機械参照閾値を超えて Content-Type とサービス状態コードの条件を満たした SubURL が参照されたということと，その SubURL が参照されたデータとリファラとして記録されたデータの間に，別の MainURL2 を参照したデータが存在するという通信履歴である．

この場合，Web ブラウジングの流れを扱うフィルタでは，SubURL を MainURL と誤認し，一方で，この SubURL の親 Node，つまり MainURL1 との参照時刻差は閲覧閾値より短い為，正解データである MainURL1 の取得漏れを起こす．

Web ページの構造に着目したフィルタでは、間に別の MainURL2 を参照しており、MainURL1 と MainURL2 の参照時刻差と MainURL2 と SubURL との参照時刻差が共に機械参照閾値より短いため、MainURL2 と SubURL を共に SubURL だと推測して排除する。一方で、MainURL1 の閲覧時間が閲覧閾値より長いと推測されたため、正解データである MainURL1 が抽出された。

E Web ブラウジングの流れを扱うフィルタでのみ抽出された誤ったデータ

この誤ったデータは、D の部分に関する考察で述べた理由で、SubURL を MainURL だと誤認し、閲覧時間が充分長かったため、抽出されたデータである。

以上の考察から、タブブラウザを利用した閲覧形態の場合の処理の遅延を考慮に入れる必要があることが分かった。

第 6 章

おわりに

本論文では、利用者の計算機利用傾向を把握する手段として、利用者の Web 閲覧履歴を把握する方式について述べた。

まず、利用者の通信履歴を記録するためのシステムとして Private Proxy Server と、Private Proxy Server の計算機利用履歴を利用する上での問題点を述べ、その対処法として、フィルタを開発することを述べた。次に、通信履歴について述べ、通信履歴と Web ページの構造の関係について述べた。そして、Web ページの構造のに着目したフィルタの設計を述べ、実験結果から判明した課題について述べた。そして、課題に対処するために、主に利用者の Web ブラウジングの流れに着目する必要があることを述べ、Web ブラウジングの流れを扱うフィルタの設計方針について述べた。Web ブラウジングの流れを扱うフィルタを実装し、Web ページの構造のに着目したフィルタに比べて、タブブラウザを利用した閲覧形態において再現率が 64 % から 91 % に向上したことを示した。

残された課題として、データ操作機能の考案や、ブラウジングツリーによる仕事の判別、過去の抽出データの利用方法の考案などがある。

謝辞

本研究を進めるにあたり，懇切丁寧なご指導をしていただきました乃村能成准教授に，深く感謝致します．

また，日頃の研究活動において，お世話になりました研究室の皆様方に，深く感謝致します．

参考文献

- [1] 大黒隼司, “個人用ネットワークサーバによる利用者支援法の検討,” 岡山大学工学部情報工学科卒業論文, 2006
- [2] “Apache module mod_log_config,”
http://httpd.apache.org/docs/1.3/mod/mod_log_config.html#formats
- [3] R . Fielding , et al, “Hypertext Transfer Protocol – HTTP/1.1,” RFC2616 , June 1999