

修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院 情報理工 学研究科 情報・通信工学 専攻 博士前期課程		
氏 名	米村 淳	学籍番号	1331115
論 文 題 目	Web アクセスにおける動的コンテンツ検索の応答性向上のための先読み機能を持ったネットワーク内キャッシュシステムの研究		
<p>要 旨</p> <p>近年、Google map や Jorudan での乗換検索，食べログの店舗検索をはじめとする動的コンテンツが多くの人に利用されている。これら動的コンテンツは，ユーザの入力や現在時刻などをもとにして，サービス提供者側のサーバで動的に生成され，ユーザに送り返される。すなわち，ユーザが要求するコンテンツを入手するために，ユーザとサービス提供者サーバは直接通信をやりとりする必要がある。このため，動的コンテンツはユーザから見た応答性を向上させることが難しい。</p> <p>本論文では，これら動的コンテンツの応答性を向上させるため，「動的コンテンツ検索先読み機能を持つネットワーク内キャッシュシステム」を提案した。本システムでは，動的コンテンツ要求をユーザに代行して先読みサーバが行い，先読みサーバ内にキャッシュとして保持し，よりユーザに近い位置でコンテンツを返送できる。提案システムを代表的な動的コンテンツの一つである乗換検索に対して実装し，その有効性を評価した。</p> <p>その結果，先読みサーバを用いることにより，両コンテンツの応答性を最短 10ms，先読みサーバの負荷が高い場合でも 100ms 以内にユーザに送信することができ，コンテンツのユーザから見た応答性を向上させた。また，最大で 85%のユーザに対してコンテンツの応答性を向上させることができた。加えて先読みサーバを複数のユーザが利用することにより，先読みサーバを用いない場合より先読み回数を削減できることができた。</p> <p>システム評価から得られたデータによって，動的コンテンツはメモリ上にキャッシュとして保持する必要があるなど，コンテンツ応答性をより向上させるための知見を得た，先読みを行う入力として，ユーザからの要求だけでなくソーシャル情報を用いる場合は，どのような状況であればソーシャル情報は有効な入力となるかを検討した。</p> <p>最後に，乗換検索だけでなくより一般的な動的コンテンツに対して本システムを適用する場合には，語の共起を利用して検索語の先読みを行う必要があるなど，考慮しなければならないことについての知見を得た。</p>			

Web アクセスにおける動的コンテンツ検索 の応答性向上のための先読み機能を持った ネットワーク内キャッシュシステムの研究

学籍番号 1331115

氏名 米村 淳

指導教員 小花 貞夫

副指導教員 寺田 実

電気通信大学大学院 情報理工学研究科 情報・通信工学専攻

2015年3月

Web アクセスにおける動的コンテンツ検索の応答性向上の ための先読み機能を持ったネットワーク内 キャッシュシステムの研究

米村 淳

概要

近年、Google map や Jorudan での乗換検索、食べログの店舗検索をはじめとする動的コンテンツが多くの人に利用されている。これら動的コンテンツは、ユーザの入力や現在時刻などをもとにして、サービス提供者側のサーバで動的に生成され、ユーザに送り返される。すなわち、ユーザが要求するコンテンツを入手するために、ユーザとサービス提供者サーバは直接通信をやりとりする必要がある。このため、動的コンテンツはユーザから見た応答性を向上させることが難しい。

本論文では、これら動的コンテンツの応答性を向上させるため、「動的コンテンツ検索先読み機能を持つネットワーク内キャッシュシステム」を提案した。本システムでは、動的コンテンツ要求をユーザに代行して先読みサーバが行い、先読みサーバ内にキャッシュとして保持し、よりユーザに近い位置でコンテンツを返送できる。提案システムを代表的な動的コンテンツの一つである乗換検索に対して実装し、その有効性を評価した。

その結果、先読みサーバを用いることにより、両コンテンツの応答性を最短 10ms、先読みサーバの負荷が高い場合でも 100ms 以内にユーザに送信することができ、コンテンツのユーザから見た応答性を向上させた。また、最大で 85% のユーザに対してコンテンツの応答性を向上させることができた。加えて先読みサーバを複数のユーザが利用することにより、先読みサーバを用いない場合より先読み回数を削減できることができた。

システム評価から得られたデータによって、動的コンテンツはメモリ上にキャッシュとして保持する必要があるなど、コンテンツ応答性をより向上させるための知見を得た。先読みを行う入力として、ユーザからの要求だけでなくソーシャル情報を用いる場合は、どのような状況であればソーシャル情報は有効な入力となるかを検討した。

最後に、乗換検索だけでなくより一般的な動的コンテンツに対して本システムを適用する場合には、語の共起を利用して検索語の先読みを行う必要があるなど、考慮しなければならないことについての知見を得た。

目次

1. 序論	1
2. 研究の背景と目的.....	3
2.1. 研究の背景	3
動的コンテンツと静的コンテンツ	3
動的コンテンツの分類	4
先読み・キャッシュの対象とする動的コンテンツ	8
コンテンツの応答性の重要性	8
モバイル環境におけるネットワークと特徴	8
2.2. 研究の目的	9
3. 関連研究	10
3.1. 静的コンテンツに関する研究	10
3.2. 動的コンテンツに関する研究	11
3.3. 動的コンテンツを対象とした関連研究のまとめと課題	14
4. システムの基本設計	15
4.1. システムの位置づけと機能概要	15
4.2. 適用する動的コンテンツ	17
5. 提案手法の実装	20
5.1. URL の分析.....	21
5.2. 先読みサーバ.....	23
Pre_search_engine	25
Social_crawler	26
5.3. 動的コンテンツキャッシュ機能	28
5.4. ユーザ入力事前送信	30
5.5. コンテンツ要求制御, データベース更新.....	31
6. システムの評価	32
6.1. システムの動作環境.....	32
6.2. システム性能評価	34
6.3. 応答性能評価 (実サービス提供者)	36
6.4. 応答性能評価 (実験用コンテンツ配信サーバ)	38
秒間問合せ回数による応答時間の変化とハードウェアリソース.....	38
6.5. ヒット率評価.....	41
1) ソーシャル情報を利用しない場合	42
2) ソーシャル情報を利用した場合	44
7. 考察	46

7.1. 評価内容に関する考察.....	46
システム性能評価に関する考察.....	46
応答性能評価（サービス提供者側サーバ）に関する考察.....	46
応答性能評価（実験用コンテンツ配信サーバ）に関する考察.....	47
ヒット率評価に関する考察.....	47
7.2. 今後の課題.....	49
8. 結論.....	52
参考文献.....	53
謝辞.....	56
付録 A 発表時資料.....	57
ネットワーク内での動的コンテンツ先読みに関する一考察.....	58
Proposal of In-Network Caching System with Dynamic Content Pre-fetching Mechanism.....	59
付録 B ソースコード.....	61
pre_search_engine.php.....	61
Social_cralwer.php.....	64
portal.php.....	69
redirector.php.....	84
付録 C 各種設定.....	87
apache の設定(httpd.conf)抜粋.....	87
apache コンパイル時設定.....	89
php コンパイル時設定.....	90
付録 D 評価用シナリオ, スクリプト等.....	91
駅間移動人員と検索回数を保持するテキスト（抜粋）.....	91
シナリオ作成用スクリプト（make_scenario.php）.....	91
ソーシャル情報を利用したシナリオクライアント側 （create_scenario_w_social_client.php）.....	92
ソーシャル情報を利用したシナリオ先読みサーバ側 （create_scenario_w_social_proxy.php）.....	94
システム性能評価用スクリプト-クライアント,先読みサーバ間(client_to_proxy.php) ..	96
システム性能評価用スクリプト-先読みサーバ,コンテンツ配信サーバ間 (proxy_to_server.php).....	98
応答性能評価-実サービス提供者サーバ用スクリプト(hittest.php).....	100
応答性能評価-実験用コンテンツ配信サーバ用スクリプト(whole_test.php).....	104

目次

図 1 ページ全体が生成される動的コンテンツ (枠内)	4
図 2 ページ生成後一部が書き換えられる動的コンテンツ (枠内)	5
図 3 期間ごと自動的に変化する動的コンテンツ (枠内)	6
図 4 URL が同一でもユーザごと内容が異なる動的コンテンツ (枠内)	7
図 5 URL が同一であればユーザが別でも内容が一致する動的コンテンツ (枠内)	7
図 6 先読みサーバの配置.....	15
図 7 システム概観.....	17
図 8 システム実装図.....	20
図 9 ポータル.....	30
図 10 システム評価時ネットワーク	33
図 11 同時接続クライアント数別の URL 生成個数と生成時間の関係 (中央値) ..	34
図 12 先読み数別の同時接続クライアント数と生成時間の関係 (中央値)	35
図 13 同時入手 URL 数とキャッシュするまでの時間の関係.....	35
図 14 Yahoo 乗換検索に対する先読み有無による応答速度の違い.....	36
図 15 Jorudan 乗換検索に対する先読み有無による応答速度の違い.....	37
図 16 秒間問合せ回数と応答速度の関係.....	38
図 17 秒間問合せ回数別, 1 時間の応答速度の変化	39
図 18 秒間問合せ回数別, 1 時間の CPU 使用率の変化.....	39
図 19 秒間問合せ回数別, ディスク書き込み量の変化.....	40
図 20 問合せ回数とキャッシュディレクトリ使用量の関係.....	40
図 21 先読み個数別, シナリオに対する先読みヒット率	43
図 22 30 個先読み時, 1 分間の先読み回数積算値.....	43
図 23 70 回先読み時, 1 分間の先読み回数積算値.....	44
図 24 書き込み数最大 Social_crawler なし.....	45
図 25 書き込み数最大 Social_crawler あり.....	45
図 26 書き込み数最小 Social_crawler なし.....	45
図 27 書き込み数最小 Social_crawler あり.....	45
図 28 降車駅数に着目した乗車駅数の累積分布.....	48

表目次

表 1	動的コンテンツを生成する URL 比較.....	17
表 2	適用するサービス	21
表 3	動的コンテンツを生成する URL 中のクエリの要素.....	22
表 4	生成する URL.....	22
表 5	テーブル (ア) WebSiteRanking, (イ)UserSearch のカラムおよび保持する要素	24
表 6	テーブル (A) Station	24
表 7	テーブル (B) Line	24
表 8	ツイート内容と内容の形態素解析.....	27
表 9	キャッシュ不可の原因.....	28
表 10	apache における cache 処理に関する設定.....	29
表 11	先読みサーバを実装したサーバ.....	32
表 12	評価用に用意したコンテンツ配信用サーバ.....	32
表 13	利用したアプリケーション.....	32
表 14	Jorudan 乗換検索における時刻別検索回数.....	42
表 15	各路線に関する 21 時から 23 時までの書きこみ時刻.....	49

1. 序論

近年、Google map[1]や Jorudan[2]での乗換検索，食べログ[3]での店舗検索をはじめとする動的コンテンツが多くの人に利用されている。文献 [4]によれば，Jorudan 乗換検索は月間で 2 億 800 万回検索されるコンテンツである。また，インターネット白書 2012[5]によれば，5639 人に行ったアンケートに対して，食べログを利用していると答えた人の率は 19.2%にのぼる。

これら動的コンテンツは，ユーザの入力や現在時刻などをもとにして，サービス提供者側のサーバで生成され，ユーザに送り返される。そのため，ユーザが要求するコンテンツを入手するには，ユーザとサービス提供者サーバは直接通信をやりとりする必要がある。

一方で，ユーザや時刻が異なってもコンテンツの内容が変化しない静的コンテンツも存在する。これらコンテンツは，内容が変化しないため，必ずしもサービス提供者側のサーバから送信する必要がなく，プロキシサーバなどよりユーザに近い位置に存在する中間サーバに存在するキャッシュから提供することが可能である。

これより，動的コンテンツは静的コンテンツと比較してユーザにより近い位置に配置することが難しく，ユーザから見た応答性を向上させることが難しい。

本論文では，これら動的コンテンツの応答性を向上させるため，「動的コンテンツ先読み機構を持つネットワーク内キャッシュシステム」を提案する。このシステムでは，ユーザの求める動的コンテンツを「先読みサーバ」で先読みしキャッシュする。これによりユーザの求める動的コンテンツをよりユーザに近い位置から提供することができ，応答時間を向上させることが可能となる。

提案するシステムの主な適用先としてモバイルネットワークを想定する。ユーザが利用するネットワークには，クライアントが直接 FTTH などにつながっている有線ネットワークと，クライアントがモバイル網などの基地局と接続している無線ネットワークが存在する。有線ネットワークの場合，ユーザの利用できる帯域には余裕があり，応答遅延も小さい。一方で無線ネットワークは，有線ネットワークと比較してユーザー一人あたりの利用できる帯域に余裕がなく，複数のユーザが均等に帯域を利用することが望ましい。また利用応答遅延も有線ネットワークと比較し大きい。一方で基地局より先は有線で接続されており，帯域に余裕がある。このため，有線無線にかかわらず通信に影響の少ない位置で先読みを行う。

しかし，すべての動的コンテンツに対して先読みサーバで先読みすることは難しい。これは，動的コンテンツを生成するための問い合わせフォーマットは，サービスによって様々

であり、また入力される検索語も多岐にわたるためである。そのため、本論文では乗換検索のサービスに関する動的コンテンツに限定して、先読み機能を実装し評価する。これは、乗換検索においては検索する際の問い合わせフォーマットに規則性があり、また、自由にユーザが入力する文字列も駅名に限定されるためである。

本論文では以下の内容について議論し、「先読み機構を持ったネットワーク内キャッシュシステム」の有効性を示す。また得られた結果をもとに、乗換検索に限らない一般的な動的コンテンツに対し、先読みを行う際の知見を述べる。

- 静的コンテンツと動的コンテンツに関する本論文における定義
- 動的コンテンツの種類と本論文において適用する動的コンテンツ
- 関連研究と課題
- 先読み機構を持ったネットワーク内キャッシュシステムの概要とその実装
- 本論文が適用する動的コンテンツに対する評価実験とその結果
- 評価実験に関する考察と一般的な動的コンテンツへの適用方法

本論文は以下の構成からなる。2節では本研究の背景となる動的コンテンツの概要やコンテンツの応答性の重要性について述べる。次いで、本研究の目的を述べる。3節では、静的コンテンツにおける先読み、キャッシュに関する従来の研究を紹介し、その手法が動的コンテンツに対して適用できないことを述べる。また、動的コンテンツに対して行われている先読み、キャッシュに関する研究を紹介し、それら研究の特徴と課題を述べ、本論文の立ち位置を示す。4節では、提案する「動的コンテンツ先読み機構を持ったネットワーク内キャッシュシステム」の概要を述べる。5節では、提案システムを実際に動的コンテンツである乗換検索を対象として実装を行い、その具体的処理を論じる。6節では、実装したシステムに関して、応答速度の向上などを評価する。7節では、6節で行った評価に対して考察を行い、より一般的な動的コンテンツに適用する場合の方法や課題を議論する。

2. 研究の背景と目的

2.1. 研究の背景

動的コンテンツと静的コンテンツ

本論文において動的コンテンツとは、内容がサービス提供者側のサーバでユーザの要求により動的に生成され、ユーザに送り返されるウェブページを指す。これらページは、PHP や Java で記述されたスクリプトによって、サービス提供者側のサーバに存在するデータベースと連携してページを作成し、ユーザに送り返される。このため、ユーザがコンテンツ要求を行うまで、コンテンツそのものはサービス提供者側のサーバにも存在しない。また、動的コンテンツを生成するためには、ユーザから検索文字列の入力などの入力を受ける必要がある。このため、コンテンツの先読みを行ってユーザの本要求より前にコンテンツを入手することが難しい。

一方、静的コンテンツはコンテンツをサービス提供者が生成してから、その内容が変更されないものである。例えば、静的な HTML ファイルや画像ファイルが静的コンテンツにあたる。この場合、コンテンツは HTML や画像としてサービス提供者側のサーバに保管されており、それをユーザの要求に応じて送信する。また、コンテンツの内容が変化しないため、サービス提供者側のサーバだけでなく、ユーザとサービス提供者の間に存在するプロキシサーバや、ユーザの利用しているブラウザなどのキャッシュに保存し、そこからコンテンツを提供することも可能である。また、静的コンテンツを指す URL も一意であるため、先読みを行ってユーザが実際に要求を行う前にコンテンツを入手しておくことも可能である。

以上より静的コンテンツと比較して、動的コンテンツは、より新しいコンテンツやユーザごとに異なる内容のコンテンツをユーザに提供することができる代わりに、

- 先読みを行ってユーザが求めるより前にコンテンツを入手すること
- コンテンツをよりユーザに近い位置から提供すること

が難しい。これらの点から動的コンテンツは、ユーザへコンテンツが到達するまでの時間を短縮できないという課題がある。

動的コンテンツの分類

2.1.1 節で一般的な動的コンテンツは、静的コンテンツと比較して先読みおよびキャッシュが難しいということについて述べた。

本節では次に、どのタイミングで動的に変化するかという観点から動的コンテンツの分類を行う。また、生成の仕方に基づいて動的コンテンツの分類を行う。さらに、分類した動的コンテンツについて先読み、およびキャッシュ導入の有効性について議論する。

Adar[6]らは、動的コンテンツを以下の3種類に分類した。

- 1) ウェブページ全体は動的に生成されるが、その後は変化しないもの
- 2) DOM (Document Object Model) によってページ生成後も書き換えられるもの
- 3) ある期間ごとに自動で書き換わるもの

1)ではページ全体がユーザの入力などによって生成される。動的コンテンツを指す URL はコンテンツが生成された時点で決定され、URL と動的コンテンツは1対1に対応付けられる。データのやり取りは、HTTP のプロトコルに則って行われる。このような動的コンテンツの例として、乗換検索が挙げられる。乗換検索では、ユーザの入力した乗車駅情報などをもとに、サービス提供者側のサーバでページ全体が生成されユーザに送り返される(図1)。再度検索を行う場合も、ページはすべて新たにサービス提供者サーバで生成され、送信される。



図1 ページ全体が生成される動的コンテンツ(枠内)

2)では、サービス提供者サーバから送信されたウェブページの内部に、Javascriptなどのクライアント側で動作する言語を記述しておく。その後ユーザの操作に応じて、表示しているコンテンツの一部を書き換える(図2)。この場合、コンテンツを書き換えるためのデータはバックグラウンドでサービス提供者サーバとやり取りされる。またやり取りするデータのフォーマットなども、サービス提供者側が自由に規定できる。このため、サービス提供者以外が制御することは難しい。

3)では、ユーザの入力にかかわらず、ある特定の時間ごとにコンテンツが更新されていく。このような動的コンテンツの例として、株価の情報を表示するようなコンテンツが挙げられる。株価は頻繁に変化していくので、コンテンツは定期的に更新される。この更新は、サービス提供者側から周期的に送信されるデータを利用して表示する(図3)。このデータはサービス提供者以外ではあらかじめ用意することはできず、またデータのフォーマットもサービス提供者の設定による。そのため、このタイプの動的コンテンツも、サービス提供者以外が制御することは難しい。

以上より、2)、3)においてはコンテンツの更新が頻繁に発生するうえ、コンテンツ書き換えのタイミングが複雑である。またコンテンツの一部のみが変化することも多い。従って、先読みやキャッシュの効果を得ることが難しい。一方1)に関しては、コンテンツの生成に関しては動的にサービス提供者側サーバで行われるが、それ以降内容は変化しない。また、動的コンテンツを指すURLもコンテンツが生成された時点で決定され、同一のURLであれば同一の動的コンテンツとなる。これより、このタイプの動的コンテンツはキャッ



図2 ページ生成後一部が書き換えられる動的コンテンツ(枠内)



図 3 期間ごとと自動的に変化する動的コンテンツ (枠内)

シュについて効果があると考えられる。また、URL を決定することができれば先読みの効果も得られると考えられる。

1)~3)では、動的なウェブコンテンツがどのように動的かという点から分類を行った。加えて、動的コンテンツはその内容から以下の二つに分類可能と考えられる。

- i) コンテンツを要求したユーザごとに異なるもの
- ii) ユーザが別でも共通となるもの

前者 i) の場合、コンテンツを要求する URL が同一でも、内容が異なる可能性がある。これは、ブラウザに保存された cookie やセッションの情報を利用してコンテンツを生成しているためである。このようなコンテンツの例として、amazon のトップページなどがあげられる (図 4)。Amazon では、トップページを要求するための URL はどのユーザによっても一意であるが、表示される内容はユーザによって異なる。よって動的コンテンツを先読みしておくことは効果があると考えられるが、キャッシュすることはできない。

後者 ii) の場合、異なるユーザが動的コンテンツを求めた場合でも、URL が同一であるならば同一のコンテンツが表示される。このようなコンテンツの例は、乗換検索や食べログなどの動的コンテンツである。これらはユーザの検索文字列によって URL が決定され、コンテンツの内容も URL に対して一定となる (図 5)。従ってこのような動的コンテンツは先読み、キャッシュともに効果があると期待できる。

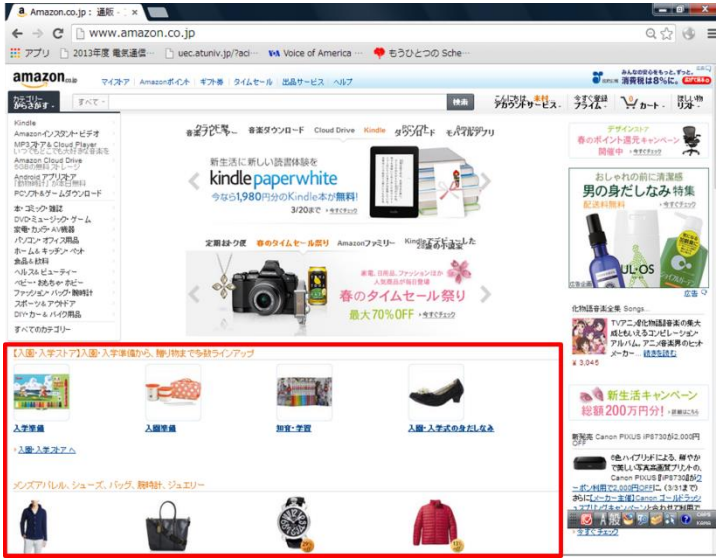


図 4 URL が同一でもユーザごと内容が異なる動的コンテンツ (枠内)



図 5 URL が同一であればユーザが別でも内容が一致する動的コンテンツ (枠内)

先読み・キャッシュの対象とする動的コンテンツ

前節までで動的コンテンツの応答性向上に向けた課題を整理した。また、応答性向上のための動的コンテンツの先読み・キャッシュの適用性の観点から動的コンテンツの分類を行った。以下、前節で行った分類に基づいて先読み・キャッシュの対象とする動的コンテンツの特性を述べる。

本論文において、対象とする動的コンテンツは以下を満たすものとする。

- 生成される動的コンテンツは前節 1) “ウェブページ全体は動的に生成されるが、その後は変化しないもの” に該当する
- コンテンツの内容は前節 ii) “ユーザが別でも共通となるもの” に該当する

ユーザの入力したクエリが HTTP の GET メソッドによってサービス提供者側サーバに到達する場合、その通信にはサービス提供者側が独自の機能を利用する余地がない。従って外部からでも制御を行いやすい。また先読みを行う場合に時刻などの情報が暗黙的に利用されることがなく、この点においても先読みを行いやすい。加えて、生成されるコンテンツがウェブページ全体となり、ユーザが別でも共通となることによって、動的コンテンツをキャッシュする効果がある。先読みとキャッシュの効果を得ることができる動的コンテンツを本論文では対象とする。

コンテンツの応答性の重要性

コンテンツがユーザに届くまでの時間は重要である。Google の調査[7]では、検索結果を返す時間が 0.5 秒遅延することにより、トラフィックが 20%低下し、結果として収益も低下すると報告されている。また Nielsen[8]は、表示までにかかる時間が 0.1 秒以内であれば「システムが瞬時に反応している」と反応、1.0 秒以内であれば「思考の流れは途切れないうが、遅延が発生していると感じる」とユーザが感じ、10 秒を超えると、ユーザは画面に対して集中力を失い、他のことを行おうとすると報告している。

以上から、コンテンツの応答性はユーザの体感に大きい影響を与え、これを短縮することによってユーザのコンテンツ利用の快適性や利便性を上げることができると考えられる。

モバイル環境におけるネットワークと特徴

ユーザの利用するクライアントまで有線で接続されているネットワークと異なり、モバイルネットワークはユーザの端末と基地局の間で電波を利用して通信を行う。また、一つの基地局に複数のユーザが接続するため、ユーザひとりが利用可能な帯域は制限がある。

また、近年モバイルネットワークのトラフィックは増加しており、モバイルネットワークを利用する通信をできる限り他のネットワークを経由させるオフローディングに関する研究も行われている[9][10][11]。従ってモバイルネットワークでは、(動的コンテンツの応答性を向上させるための処理の中の)先読み処理をユーザの所持するクライアントで行うべきではない。一方で、基地局よりサーバ提供者側に近い事業者ネットワークは、基本的には光ファイバーや同軸ケーブルなどを用いた有線接続であり、無線ネットワークと比較し帯域に余裕がある[12]。

従って、より帯域に余裕のある基地局より先の事業者ネットワーク内で先読み処理を行うことで、基地局とユーザの間の帯域を圧迫せず、ユーザの求める動的コンテンツを、よりユーザに近い位置で用意可能となる。

2.2. 研究の目的

本論文は、静的コンテンツと比較して応答性の向上が難しい動的コンテンツについて、その応答性を向上させ、ユーザの利便性を上げることを目的とする。応答性向上のため、先読み機構を持ったネットワーク内キャッシュシステムを提案し、乗換検索に関する動的コンテンツに対して実装、評価、考察する。また、一般的な動的コンテンツに対し、提案する先読み処理がどのように適用可能かについても考察する。

3. 関連研究

動的コンテンツの応答性向上については、様々な研究がおこなわれている。静的、動的に限らずコンテンツを早く入手する手法は次の二種類に大別できる。

- A) キャッシュを行うもの
- B) 先読みを行うもの

また、これらの手法を 1)クライアント側で行うか、2)サービス提供者側で行うか 3)プロキシサーバなど中間サーバで行うかに分けることができる。

本節ではまず、ユーザの入力によって内容の変化しない静的コンテンツに関する研究について述べ、それがなぜ動的コンテンツに適用できないかを論じる。次に、動的コンテンツに対する研究を述べ、その問題点を論じる。

3.1. 静的コンテンツに関する研究

静的コンテンツをキャッシュする手法については、クライアント側では、一般的なブラウザに静的コンテンツをキャッシュする機能が実装されている[13]。クライアント側では、一度アクセスした静的コンテンツを保持し、再度同一クライアントが要求した場合はキャッシュに保持しているコンテンツを表示する。この場合、2回目以降の表示速度は改善するが、初回のアクセスは速度向上が見込めない。また、キャッシュに保持したコンテンツはクライアントを使用するユーザのみしか利用できない。

また、サービス提供者側においては、一般的なサーバアプリケーションの一部として、静的コンテンツをキャッシュする機能が実装されている[14]。サービス提供者サーバでは、ユーザからのコンテンツ要求を処理するウェブサーバのほか、サーバの機能としてキャッシュを利用し、ウェブサーバの負荷を抑えコンテンツを提供可能である。しかしサービス提供者側までの通信は必ず発生するため、サービス提供者サーバ内での応答遅延が抑制されるのみであり、ユーザとサービス提供者サーバ間の応答遅延短縮効果は薄い。

中間サーバにおけるキャッシュ[15]でも、あくまで一度受信したコンテンツを保持し、ユーザの要求に応じて送信する。中間サーバでのキャッシュに関しては、効率の良いキャッシュ置換に関する研究が主である。中間サーバにキャッシュを保持することによって、同一の静的コンテンツ要求が複数のユーザから来た場合はそのコンテンツを複数のユーザで再利用することが可能である。しかし、この場合も最初に要求を行ったユーザに関して

はサービス提供者サーバまで問い合わせを行う必要があり、応答遅延短縮効果はない。

一方、先読みに関しては、クライアント側・中間サーバ・サービス提供者側での提供を想定した様々な研究が行われている。Sundaresan らの研究[16]では、クライアントに近いホームネットワークにおいて、ユーザの要求が多いドメインに対して先読みを行いキャッシュする Popularity Based Prefetching を提案している。この手法では、よく利用されるサイトのドメインを DNS プリフェッチで入手し、静的コンテンツをキャッシュする。

また Yeh らの研究[17]では、プロキシサーバにおいて先読みを行う際に Group Prefetching という手法を用いて先読みを行っている。この手法は、最近アクセスされたコンテンツと、URL に示されているディレクトリ構造の近さを用いて、先読みする対象を決定する。この手法でも、ドメインに基づく先読みが行われる。

Teng らの研究[18]では、プロキシサーバにおけるキャッシュ容量を考慮したキャッシュ・先読みアルゴリズムである IWCP を提案している。この手法では、先読みに関してはユーザの閲覧履歴を利用して、ウェブオブジェクトのハイパーリンクからどのコンテンツを先読みしキャッシュするかを決定する。

Ossa らの研究[19]では、サービス提供者側においてユーザのアクセス履歴を分析し、先読みを行い、先読み結果をユーザに通知する。この手法においても先読みを行う対象は HTML の Link タグに指定されたアドレスに対して先読みを行う。

以上に示すような静的コンテンツに関する研究では、先読みには URL のドメインや、コンテンツ中のハイパーリンクを利用している。これらは、サービス提供者が構成を変更しない限り変わらない。これらの情報とユーザのコンテンツの閲覧履歴やどのページがよく閲覧されているかを組み合わせて、先読み先を決定している。一方で動的コンテンツは、ドメインのみならず、ユーザがどのようなクエリを入力するかによって、表示される内容が変化する。従って、静的コンテンツに対して適用されている手法は、動的コンテンツに適用できない。

3.2. 動的コンテンツに関する研究

3.1 節で述べたように、静的コンテンツにおける先読み手法を動的コンテンツにそのまま適用することはできない。また、動的コンテンツは基本的にサービス提供者サーバで都度生成されるため、キャッシュ手法についても静的コンテンツに適用した手法は利用が難しい。

動的コンテンツの応答性を向上させるための研究も種々行われている。まず動的コンテンツのキャッシュに関する研究を述べる。庄野ら[20]が提案する手法では、ユーザ側とサー

ビス提供者側両方にキャッシュ用のプロキシサーバを配置し、動的コンテンツでも内容が同一のものや、類似しているもののキャッシュを行い、コンテンツ表示速度を向上させ通信量も削減している。しかしこの手法ではクライアント側とサービス提供者側両方にシステムを導入する必要がありコストがかかるうえ、先読みは行わないため、コンテンツがキャッシュされていない場合は速度向上を行うことができない。

Chen らの研究[21]や Ma らの研究[22]では、動的コンテンツの中身を変化する部分と変化しない部分に分け、変化しない部分についてプロキシサーバにおいてキャッシュを行い転送量の削減とコンテンツ入手までの時間を短縮している。しかし、動的コンテンツにおいてユーザが求めるものは内容が変化する部分のコンテンツであると考えられるが、これら研究では、ユーザが求める部分の入手までの時間を短縮することはできない。また庄野らの研究と同様に先読みは行われなかったため、最も初めにアクセスを行うユーザに対する応答性改善は見込めない。

Zhang の研究[23]においても動的コンテンツを共有可能な断片 **Shared Fragments** に分割し、内容の変化しない部分についてプロキシサーバでキャッシュしている。**Shared Fragments** の構築は一度動的コンテンツがプロキシサーバに入力された段階で行われるため、この手法でも最も初めにアクセスを行うユーザに対しては動的コンテンツの応答性改善が見込めない。

また、Chang らの研究[24]では、サービス提供者側のサーバで動的コンテンツをキャッシュする手法を研究した。この研究では、ユーザからのクエリによって生成される動的コンテンツの中でも、とりわけユーザによって内容が変わるパーソナライズされた動的コンテンツについて、サーバ側でいかにキャッシュを行うかを述べている。この研究は、よりユーザの情報に左右される動的コンテンツに関して対処している点において本論文と異なる。また、ユーザから遠いサービス提供者側サーバでキャッシュするため、応答性向上効果が限定される。

以上のように、動的コンテンツのキャッシュに関する研究は、クライアント側、プロキシサーバ側、サービス提供者側のいずれでも行われている。しかしクライアント側、プロキシ側では、少なくとも一度データが取得されていなければ、動的コンテンツの応答性を上げることはできない。また、クライアント側では、キャッシュした動的コンテンツを利用できるものは要求を行ったユーザに限られる。ユーザが別でも共通となる動的コンテンツにおいては、動的コンテンツの再利用はできない。一方サービス提供者サーバ側では、要求は必ずサービス提供者サーバまで到達する必要があり、応答速度の改善は動的コンテ

ンツがスクリプトにより生成される分のみの改善に留まる。いずれの場合も、動的コンテンツのキャッシュを行うのみでは応答速度改善は限定的となる。

動的コンテンツのキャッシュに対して、動的コンテンツの先読みに関する研究も行われている。

柴田らの研究[25]では動的コンテンツを静的な部分と動的な部分に分割し、静的な部分を事前にブラウザで先読みし、動的生成される部分が生成され次第静的な部分と結合する。これによりロード時間が 20%短縮されたが、この手法ではユーザが求める動的に生成される部分については先読みされない。

Lymberououlos らの研究[26]では、クライアント上でユーザの動的コンテンツへのアクセス履歴を学習し、クライアント側が動的コンテンツを先読みし入手する。この研究においては、動的コンテンツを静的な部分と動的な部分に限らず先読みし入手するため、他の研究と異なりユーザが求める動的な部分も応答性が改善する。しかしこの手法においても動的コンテンツの再利用ができない。また、クライアント上でアクセス履歴の学習と先読みを行うため、クライアントがバッテリーなどのリソースが限られているモバイルデバイスの場合手法の適用に限りがあると考えられる。さらに、先読み結果はすべてクライアントまで到達するため、デバイスのデータ受信量も多い。

Dahlan らの研究[27]は、クライアント側で先読み対象を決定する APF (Asynchronous Predictive Fetch) と呼ばれる手法を提案した。この手法ではマウスオーバやチェックボックスのチェックなどを先読みのきっかけとして入手し、先読みする。チェックボックスの情報などが要求に含まれる動的コンテンツに対しては適用可能であるが、ユーザの自由記述入力に関しては対応できない。またこの手法も先読みはクライアント側で行うため、先読みしたが利用されなかった場合の動的コンテンツもクライアントに送信されるため、データ受信量も増加する。

Challenger らの研究[28]では、サービス提供者側サーバにおいて動的コンテンツのキャッシュと先読みを行っている。この手法では、動的コンテンツがサービス提供者側で更新された場合は、キャッシュに保持されている動的コンテンツも更新する。この手法において、動的コンテンツを提供するサービス提供者側サーバの負荷を下げ、サービス提供者内での応答性を向上させることはできるが、ユーザは動的コンテンツを要求する際は必ずサービス提供者側へ問い合わせなければならない。

3.3. 動的コンテンツを対象とした関連研究のまとめと課題

以上関連研究をまず静的コンテンツと動的コンテンツに分け、静的コンテンツに関する手法が何故適用できないかを論じた。またコンテンツの応答性を改善するための手法をキャッシュと先読みに分けて論じた。さらにその手法をネットワークのどの位置で適用するかに関しても分けて論じた。動的コンテンツに関する各手法の利点、課題をまとめると以下のようなになる。

動的コンテンツをキャッシュする手法に関しては、動的コンテンツを変化する部分と変化しない部分に分けてキャッシュを行う研究が主となる。この手法を用いた場合、クライアント側、中間サーバ側ではよりユーザに近い位置で動的コンテンツを返却できるため、応答性の向上が可能である。しかしキャッシュ手法のみを適用した場合、キャッシュが存在しない初回の問い合わせを行ったユーザに対しては、キャッシュが存在せず応答性向上が見込めない。また、サービス提供者サーバでキャッシュを行った場合、短縮される応答時間は動的コンテンツを生成するのにかかる時間のみに限定される。

動的コンテンツを先読みする手法に関しては、クライアント側で行う場合はユーザに合わせた先読みが可能である。しかし先読み結果の利用はクライアントを所持するユーザに限られ、クライアントがモバイル端末の場合は、通信がクライアントまで到達するため、トラフィックを圧迫する。サービス提供者側で先読みを行った場合、サービス提供者サーバ内の応答性向上に限られる。既存研究の分析から以下の結論が導き出される。中間サーバで行う場合、先読みした動的コンテンツをキャッシュに保持すれば動的コンテンツの再利用が可能で、よりユーザに近い位置からコンテンツを提供できるため、応答性の向上も可能である。

サーバを利用する複数のユーザで共有可能なため、同一の要求が別のユーザから先読みサーバに来た場合もキャッシュサーバまでの通信で済む。

先読みサーバはネットワーク内に存在するプロキシサーバとして実装し、先読み機能を利用したいユーザはすべてこの先読みサーバを経由してサービス提供者サーバを利用するとする。この先読みサーバは、以下の二つの機能からなる。

- 1) 先読み機能：ユーザの求める動的コンテンツを、ユーザに代行して先行取得
- 2) キャッシュ機能：先読みして取得した動的コンテンツや、以前取得した動的コンテンツを保存

先読み機能においては、ユーザが動的コンテンツを要求するための検索ボタンなどを押す前に、ユーザの入力内容を事前に取得し、動的コンテンツを先行取得に利用する。Google 検索などではユーザが検索窓にキーボードから文字列を入力する際に、打鍵後すぐに検索候補が表示される。このように、ユーザが検索ボタンを押さなくても入力した内容がサーバへ届く処理を事前送信とここでは呼ぶこととするが、本論文では、一文字ごとの事前送信ではなく、入力した単語が確定した時点で先読みサーバに入力の事前送信を行うものとする。入力された 1 文字から検索語を推定することは、本論文の主目的ではない。そのため、事前送信は入力される単語が確定してから送信されるものとする。

先読みサーバに事前送信が入ってきた場合、先読みサーバではスクリプトを用いて動的コンテンツを指定するための URL を生成する。URL の生成は、ユーザから事前送信された入力のみでは動的コンテンツの URL を生成するのに不足している場合は、先読みサーバにあるデータベースを検索してその不足分を埋め、URL を生成する。

また、先読みを行う入力としては、Twitter などのソーシャルメディアも利用する。

URL を生成したのちは、先読みサーバからサービス提供者側のサーバにコンテンツ要求を HTTP で行う。その結果、動的コンテンツがサービス提供者側サーバから帰ってくるため、その結果を先読みサーバにキャッシュとして保持する。この機能はキャッシュ機能で行う。

キャッシュ機能では、先読みによって入手した動的コンテンツや、以前に入手していた動的コンテンツを保持する。一般的にコンテンツをキャッシュするソフトウェアでは、コンテンツの URL をもとにハッシュ関数を用いてハッシュ化し、その値を利用してキーを生成する。またこのキーを利用してキャッシュの探索を行う。従って、本システムでもキャッシュ機能はコンテンツを指す URL をもとにしてハッシュ関数を用いてキーを生成し、記

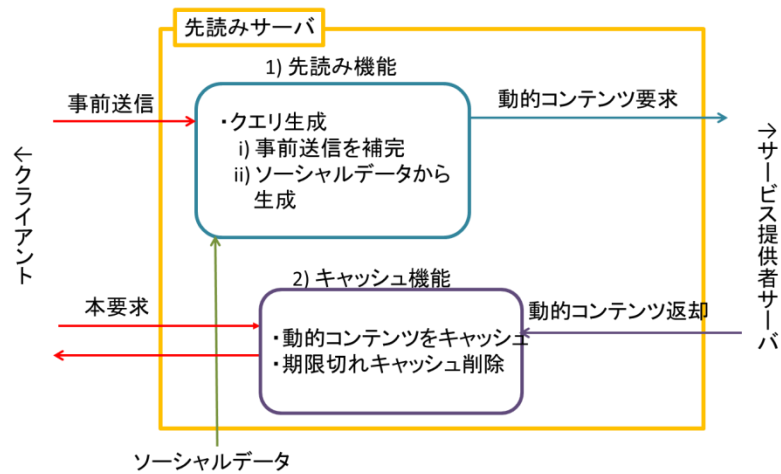


図 7 システム概観

憶領域に保持する。動的コンテンツを指す URL が一致していた場合、その動的コンテンツがキャッシュされているならばキャッシュからユーザーに送信することができる。以上の処理の流れの概要を図 7 に示す。

4.2. 適用する動的コンテンツ

前述した先読み機能を、すべての動的コンテンツに適用することは難しい。これは、動的コンテンツの生成のための問い合わせフォーマットがサービスによって様々だからである。

例として、Yahoo 乗換検索と食べログを挙げて説明をする。Yahoo 乗換検索の問い合わせ URL と食べログの問い合わせ URL は以下の表 1 に示すものとなる。

表 1 動的コンテンツを生成する URL 比較

Yahoo 乗換検索	http://transit.loco.yahoo.co.jp/search/result?from= 調布駅 &via=&via=&via=&to= 東京駅 &yym=201405&d=23&hh=10&m2=2&m1=1&al=1&shin=1&ex=1&hb=1&lb=1&sr=1&s=0 &expkind=1&ws=2&ticket=ic
食べログ	http://tabelog.com/rst/rstsearch/?LstKind=1&voluntary_search=1&lid=top_navi1 &sa= 新 宿 駅 &sk= 居 酒 屋 &x=21&y=13&area_datatype=RailroadStation&area_id=5172&key_datatype=&key_id=

これら二つの動的コンテンツは、

- ユーザの入力した検索語（クエリ）によって内容が決定される
- クエリは HTTP の GET によってサービス提供者サーバに送信される
- 生成される動的コンテンツはページ全体が生成され、ユーザに送られたのちは変化しない
- コンテンツの内容はユーザが別でも共通となる

となり、ともに本論文で対象とする動的コンテンツを満たす。しかし、その動的コンテンツを生成するための URL は大きく異なる。

Yahoo 乗換検索の場合は、乗車駅と降車駅、経由駅が自由入力（表 1 の Yahoo 乗換検索網掛け部）となり、その他の日時や時刻、交通機関の利用などは複数の候補からの値の選択となる（表 1 の Yahoo 乗換検索枠内）ため、入力される値が制限される。加えて自由入力欄も、入力される可能性のあるものは駅名に限られ、入力内容の予測が行いやすい。一方食べログの場合、自由入力となるのは地域と店のタイプ（表 1 の食べログ枠内）となる。地域や店のタイプの組み合わせは膨大となり、また予測も行い難い。また、サービス提供者が一意に設定している ID の情報（表 1 の食べログ網掛け部）なども URL に含まれてしまっているため、外部から手を加えることが難しい。

上記問題により、すべての動的コンテンツに適応可能な先読み機構を実現することは難しい。本論文ではまず、乗換検索に動的コンテンツを絞り、先読み機能とキャッシュ機能を実装する。

乗換検索を対象とする理由は、以下の 2 点による。

- 入力内容の予測を立てやすい
- 乗換検索は現在多くの人に利用されており、運行情報のソーシャルメディアへの書き込みも多く行われている。

乗換検索は、自由に入力される単語は基本的には駅名に限定される。そのため、ユーザの入力を事前入手した際に、ユーザが今後入力すると考えられる内容を類推しやすい。また駅名などの情報はデータベースとして公開されている[30]ため、先読みを行う際に候補を十分に用意可能である。

また、乗換検索は多くの人に利用されている。文献[5]の通り、ジョルダン株式会社の運営する乗換案内だけをとっても、月間の検索回数は 2 億 800 万回に達する。これは一日に

約700万回の検索が行われていることを意味する。このように乗換検索は非常に多くの人々に利用されている動的コンテンツであると言える。

さらに乗換検索を行うきっかけになると考えられる交通機関の運行情報は、ソーシャルメディアに書き込まれやすい[31]。ユーザの書き込みをもとにして、運行情報をリアルタイムで広報する「ジョルダンライブ」というサービスも存在[32]する。このサービスではTwitterに書き込まれた情報をもとに、各鉄道路線の運行情報をリアルタイムで更新していくものである。この情報はTwitterに書き込まれるため、ソーシャルメディアを利用したサービスであるといえる。

このように、乗換検索は先読みを行いやすく、また多くのユーザが利用しているため先読みによる応答時間短縮の恩恵を受けるユーザは多いと考えられる。このため本論文では、一般的な動的コンテンツに対しての先読み処理を考える前に、乗換検索に関して提案手法を実装する。

本システムにおいて先読み機能を実装する乗換案内サービスは、以下の2つとした。

- Yahoo 乗換検索
- Jorudan 乗換検索

これらサービスは乗換検索として有名であり、また本論文が対象としている動的コンテンツの要件を満たす。

5. 提案手法の実装

前節で、本システムの概要と適用する動的コンテンツについて述べた。本節では、システムの実装について述べる[33]。本システムの実装は、以下の5点からなる。

1. 乗換検索に関する動的コンテンツを生成する URL の分析
2. 先読みサーバの実装
3. 動的コンテンツキャッシュ機能の実装
4. ユーザの入力の事前送信に関する実装
5. データベース更新, 新規乗降駅挿入, およびコンテンツ要求制御に関する実装

動的コンテンツを生成する URL のフォーマットを解析することにより、先読みを行う際にどのように URL を生成すればよいかを求める。解析した情報をもとに、ユーザの求めている動的コンテンツを先読みする部分である「先読みサーバ」の実装を行う。また、ユーザの入力を事前送信することが必要であり、この部分に関する実装を述べる。加えて、先読みを成功させるには乗降駅の組を増加させ先読み候補を増やしていく必要がある。また、システムに存在しない先読み候補をデータベースに保存する必要がある。最後にこの部分に関する実装を述べる。

実装したシステムを図 8 に示す。

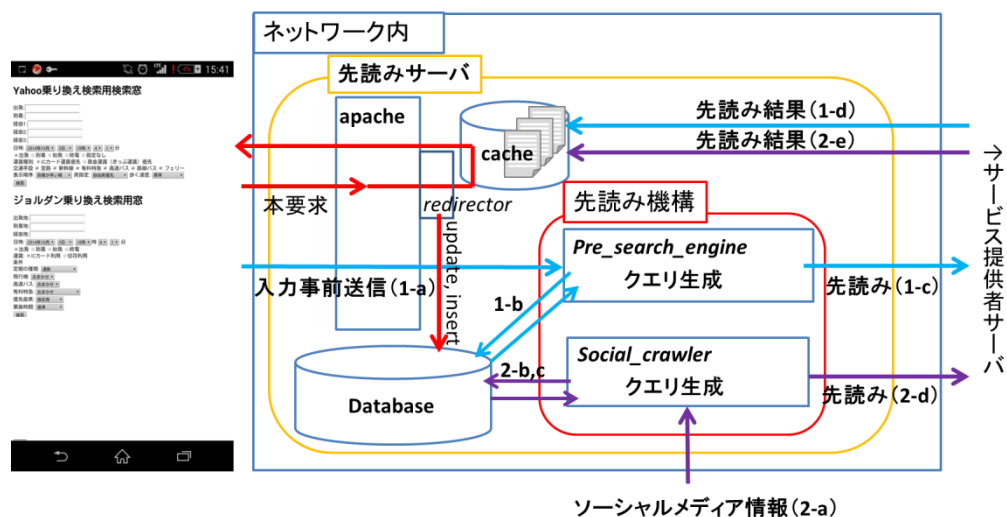


図 8 システム実装図

5.1. URL の分析

本論文で対象とする乗換検索である Yahoo 乗換検索, Jorudan 乗換検索の各サービスの動的コンテンツは HTTP の GET メソッドで要求が行われる。従って, ユーザの入力したクエリ Q は URL にじかに記述され, そのフォーマットはそれぞれのサービスにより表 2 に示す通りとなる。

表 2 適用するサービス

サービス	URL
yahoo 乗換検索	<code>http://transit.loco.yahoo.co.jp/search/result?Q</code>
jorudan 乗換検索	<code>http://www.jorudan.co.jp/norikae/cgi/nori.cgi?Q</code>

すべてのサービスで URL は `scheme//host/path?Q` の形となっている。また Q について分解すると, 動的コンテンツを要求するのに必要な要素は以下に示す表 3 のとおりとなる。

それぞれのサービスにおけるクエリを構成する要素名は異なるが, その中に格納される値は共通性がある(表 3)。

自由記述となって入力されるものは駅名のみである。一方で, 駅名以外の日時情報やどの交通機関を利用するかの入力は, HTML のセレクトボタンやラジオボタンで入力が制限される。従ってこれら乗換検索においては, 先読みをする場合は駅名について補完を行えばよい。駅名の補完は, 乗車駅が入力された場合は降車駅を補完する。

また, URL で記述される要素名の並びは, どのユーザが要求を行ってもサービスごとに統一する必要がある。本システムでは, サービス毎表 4 に示す URL を生成するようにした。Jorudan に関しては, ユーザが入力するもののほかに, ユーザごと変化しないクエリも存在する。このクエリについても, 表 4 に示す URL には含まれる。

表 3 動的コンテンツを生成する URL 中のクエリの要素

yahoo 乗換検索		jorudan 乗換検索	
要素名	値	要素名	値
From	乗車駅	eki1	乗車駅
To	降車駅	eki2	降車駅
Via	経由駅	eki3	経由駅
Ym	年月	Dym	年月
D	日	Ddd	日
Hh	時間	Dhh	時間
m1	分(10 の位)	Dmn1	分(10 の位)
m2	分(1 の位)	Dmn2	分(1 の位)
Type	検索タイプ	Cway	検索タイプ
Ticket	券種	Cfp	券種
Al	飛行機利用	C7	定期種類
Shin	新幹線利用	C2	飛行機利用
Ex	特急利用	C3	高速バス利用
Hb	高速バス利用	C1	特急利用
Lb	路線バス利用	C4	優先席種
Sr	フェリー利用	C6	乗換時間
S	表示ソート		
Expkind	席指定		
Ws	歩行速度		
Kw	降車駅		

表 4 生成する URL

Yahoo	http://transit.loco.yahoo.co.jp/search/result?from=&to=&via=&via=&via=&ym=&d=&hh=&m1=&m2=&type=&ticket=&al=&shin=&ex=&hb=&lb=&sr=&s=&expkind=&ws=&kw=
Jorudan	http://www.jorudan.co.jp/norikae/cgi/nori.cgi?eki1=&eki2=&eki3=&via_on=&Dym=&Ddd=&Dhh=&Dmn1=&Dmn2=&Cway&Cfp=&C7=&C2=&C3=&C1=&C4=&C6=&S=%E6%A4%9C%E7%B4%A2&Cmap1=&rf=nr&pg=0&eok1=&eok2=&eok3=&Csg=1

5.2. 先読みサーバ

対象とするサービスの URL を解析したことにより、先読みサーバでどのように URL を生成し、先読みすればよいかを判明した。本節では先読みサーバの実装について述べる。

先読みサーバ全体は、`apache` を用いたサーバとして実装した。この先読みサーバは、ユーザからの事前入力やソーシャル情報を入手して URL を作成するサーバとしての機能と、ユーザの要求を代行するプロキシサーバの機能を持つ。また、先読みを行うためのスクリプトは `php` で実装した。先読みを行う際のユーザの入力を推定し補完するためのデータベースも `MySQL` を利用して用意した。先読み機能は、次に示す `php` で記述した二つのスクリプトによって実現する。

- A) `Pre_search_engine`
- B) `Social_crawler`

`Pre_search_engine` は、ユーザからの事前入力送信されてきた場合に、動的コンテンツを要求するための URL を生成し、先読みを行うスクリプトである。`Social_crawler` は、`twitter` のソーシャル情報を利用して定期的に先読みを行うためのスクリプトである。

両スクリプトも、先読みを行うために駅名を補完する必要がある。そのための駅名データを保存するデータベースを実装した。データベースは大きく分けて以下の 3 つのテーブルを用意した。

- (ア) 乗換ランキング(`WebSiteRanking`)
- (イ) ユーザ検索履歴(`UserSearch`)
- (ウ) 駅・路線データ

乗換ランキングは、サービス提供者が公開している駅間の検索ランキングをテーブルに持つ。またユーザ検索履歴は、乗換ランキングに存在していない乗降駅間の検索がシステムに入力されてきた場合に、乗降駅の組を保持し次回以降の先読みを利用する。これらテーブルは `Pre_search_engine`、`Social_crawler` とともに利用する。この 2 つのテーブルは表 5 に示すデータをカラムに持つ。(ウ) 駅・路線データは、駅名と路線名、またどの駅がどの路線に属しているかを保持するテーブル群である。このデータは、

A) 駅名テーブル(Station)

B) 路線テーブル(Line)

の2種類のテーブルで構成され、それぞれは以下の表6、表7に示すデータを保持する。このテーブルは、Social_crawlerによって利用される。このテーブルを作成するに当たり、駅データ.jp[30]の提供している駅名、路線名などのデータを利用した。

以上のデータベースを利用し、先読みを行いユーザに代行して動的コンテンツを取得する。

以下より各スクリプト内の処理について述べる。

表5 テーブル (ア) WebSiteRanking, (イ)UserSearch のカラムおよび保持する要素

カラム名	保持する要素
Id	ID
Src	乗車駅
Dst	降車駅
Rank	検索ランキング公開ランク
Times	先読み成功回数
last_searched	最新検索時間

表6 テーブル (A) Station

カラム名	保持する要素
Id	駅固有 ID
Name	駅名
line_id	駅が属する路線の ID
src_popularity	出発駅として選ばれた回数
dst_popularity	到着駅として選ばれた回数

表7 テーブル (B) Line

カラム名	保持する要素
Id	路線固有 ID
Name	駅名
line_id	駅名 (カタカナ)

Pre_search_engine

Pre_search_engine では、ユーザの入力の事前送信を利用して動的コンテンツの先読みを行う。本システムにおいて、入力がユーザから事前送信されるタイミングは以下のタイミングとした。

- A) 乗車駅が入力されたとき
- B) 降車駅も入力されたとき
- C) 日時や利用交通機関が変更されたとき

上記のタイミングでユーザからの事前送信が発生し、その際にユーザが入力していた値をパラメータとして用い、Pre_search_engine が起動し先読みを行う。

pre_search_engine は、乗降駅の組が入力されたタイミングではなく、乗車駅が入力された段階で先読みを行う。これは、乗車駅と降車駅の入力のタイムラグを利用して先読みを行うためである。乗車駅と降車駅が入力された段階で先読みを行う場合、先読みに行える時間はユーザが降車駅を入力し終わり検索ボタンを押すまでの時間に限られる。一方で乗車駅が入力された時点で先読みを行った場合、降車駅を入力する時間を利用して先読みを行うことが可能である。このため、乗車駅が入力された時点で先読みを行うことで、先読みサーバにおいてユーザが本要求を行う前にコンテンツを用意することが可能になる。処理の流れの概要を以下に示す。

- 1-a) 先読みサーバが、ユーザのクライアントから入力の事前送信を受信
- 1-b) Pre_search_engine を起動して、動的コンテンツ要求を行うために必要なクエリを、乗換ランキングとユーザ検索履歴テーブルを利用して補完
- 1-c) 動的コンテンツ要求を行うことのできる URL を生成してサービス提供者サーバに HTTP の GET メソッドで送信
- 1-d) サービス提供者サーバから帰ってきた動的コンテンツをキャッシュに保持

Pre_search_engine が行う先読み処理は、入力の事前送信タイミング A) ~ C) によって処理が異なる。A) の場合、コンテンツ要求のための URL を生成するために降車駅を補完する必要がある。降車駅以外の時刻や利用する交通機関などの入力に関しては、ラジオボタンなどによって既に入力されている。この場合、以下に示す処理を適用して先読み用の URL を生成する。

1. ユーザの入力から先読み要求から乗車駅 `src` を取得
2. データベース `WebSiteRanking` とデータベースと `UserSearch` の `src` カラムを検索し、降車駅の候補 `dst` を取得.
3. `src` と `dst` の組をもととして先読みする乗降駅を決定
4. 先読みを利用した `src`, `dst` の組の `last_searched` カラムの値を現在時刻に更新
5. ユーザの入力を利用して時刻, 検索タイプ, 利用交通機関, ソートなどの情報を補完
6. 先読みで問合せる URL を生成.

B の場合, すべての入力欄は埋まっている. この場合はユーザからの事前入力に従って先読みを行う.

C の場合, 降車駅が入力されている場合とされていない場合の 2 つの可能性がある. 入力されていない場合は A の処理を, 入力されていた場合は B の処理を行う.

Social_crawler

`social_crawler` では, `Twitter` の書き込み情報を用いて乗換検索を先読みする. 先読みには `ジョルダンライブ` に書き込まれている路線の遅延情報を利用する. `Social_crawler` の処理の概要は以下に示すとおりである.

- 2-a) `Twitter` のつぶやきを周期的に収集
- 2-b) 駅・路線の運転見合わせなどの情報を抽出
- 2-c) 路線に関連する駅を駅・路線データから入手し, さらにランキング, ユーザ入力履歴と連携して問い合わせ (クエリ) を生成
- 2-d) サービス提供者側の `Web` サーバへ `HTTP` の `GET` メソッドで送信
返却された動的コンテンツをキャッシュに保持

`Social_crawler` では, `ジョルダンライブ` の情報を毎分 1 回検索する. 検索には `TwitterAPI`[34]の中の `REST API`[35]に含まれる `GET` によってツイートを検索する `API` を利用する. この `API` を利用することにより, 検索結果を `json` 形式で入手することができる. `json` をパースすることにより, ツイートを入手する.

書き込みは表 8 のテキストに示すように, テキストの状態でも入手することになるため, 書き込みから路線名を抽出する必要がある. この抽出を行うため, 日本語の形態素解析を行うことができる `Mecab`[36]を利用した. また, 路線名を形態素解析で抽出するために, 駅データ.jp より得られた路線名データベースを利用して, 路線名に関する辞書を作成した.

テキストで取得したツイートは Mecab を利用することで表 8 の形態素解析後に示すように単語ごとに分割できる。この分割した単語から、路線名を取得する。

表 8 ツイート内容と内容の形態素解析

テキスト	#jordainライブ [01/27 12:57] #山手線 [田町(東京)～渋谷] 運転再開 田町駅発車時にホームドアに異物が挟まるトラブル。 http://t.co/zkxKxF0z7lArray
形態素解析後	<p>...</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">名詞,固有名詞,地域,一般,*,*,山手線,ヤマノテセン,ヤマノテセン,1</div> <p>記号,括弧開,*,*,*, [, [, [名詞,固有名詞,地域,一般,*,*,田町,,2 記号,括弧開,*,*,*, (, (, (名詞,固有名詞,地域,一般,*,*,東京,,2 記号,括弧閉,*,*,*,) ,) ,) 記号,一般,*,*,*,~,~,~ 名詞,固有名詞,地域,一般,*,*,渋谷,,2 ...</p>

上記のように路線名を入手したのち、以下に示す手順によって、先読みを行う候補を選ぶ。

1. 路線名を利用して Line データベースから id を検索。
2. 取得した id を利用して Station データベースを検索し、路線に属する駅を src_popularity の上位から乗車駅 src を選択
3. 得られた src を利用して WebSiteRanking データベースと UserSearch データベースを検索し、times と last_searched の上位から降車駅 dst を選択
4. 選択した src, dst の組の last_searched カラムの値を現在時刻に更新
5. 時刻は現在時刻を、利用交通機関などはポータル（後述）がユーザに提供する検索の初期状態と同様な値を用いて他のクエリを補完
6. 先読みで利用する URL を生成

5.3. 動的コンテンツキャッシュ機能

前節の Pre_search_engine と Social_crawler で先読みを行った動的コンテンツは、先読みサーバからユーザへ提供するためにキャッシュに保持する必要がある。キャッシュ機能は、apache のモジュールである mod_cache[37]を利用する。

ウェブコンテンツのキャッシュは、RFC2616[38]によって規定されている。この規定と apache のキャッシュ機能を合わせると、モジュール mod_cache はキャッシュ可否を以下の手順で判断する[39]。

- 1) apache においてキャッシュ可能 URL に指定されている
- 2) レスポンスの HTTP ステータスは 200, 203, 300, 301, 410
- 3) リクエストは GET で行われる
- 4) レスポンスに Authentication ヘッダが存在し、Cache-Control ヘッダに s-maxage,max-revalidate,public オプションが指定されている必要がある。これ以外の場合、キャッシュしない
- 5) URL にクエリストリングが存在するとき、レスポンスが明確な期限切れを Expires で指定してない場合、もしくは Cache-Control で max-age, s-maxage を指定していない場合、キャッシュしない
- 6) レスポンスが HTTP ステータスコード 200 のとき、Etag が Last-modified か Expires を持つ必要がある
- 7) Cache-Control が private の時、apache において CacheStorePrivate という設定が有効になっていない限りキャッシュしない
- 8) Cache-Control が no-store の時、apache において CacheStoreNoStore という設定が有効になっていない限りキャッシュしない
- 9) レスポンスの Vary ヘッダに Match-all “*”が存在する場合キャッシュしない

乗換検索の動的コンテンツにおいては、上記手順のうち 4)以降でキャッシュ可能の条件を満たさない(表 9)。

表 9 キャッシュ不可の原因

乗換検索サービス	キャッシュ不可原因
Yahoo	Expires が存在しない Cache-Control が private
Jorudan	Expires が存在しない Cache-Control が存在しない

従って動的コンテンツをキャッシュするには apache の設定を必要とする。本システムでは表 10 の設定を行い、動的コンテンツをキャッシュする。

表 10 apache における cache 処理に関する設定

```
CacheEnable disk http://transit.loco.yahoo.co.jp/
CacheEnable disk http://www.jorudan.co.jp/
CacheIgnoreCacheControl on
CacheIgnoreNoLastmod on
CacheStoreNoStore on
CacheStorePrivate on
CacheIgnoreHeaders Set-Cookie User-Agent

Header set Expires "Wed, 01 Apr 2015 00:00:00 GMT"
Header unset Vary
```

サービス提供側サーバから送信される動的コンテンツのキャッシュへの保持を許可するため、CacheEnable を設定する。これにより、上記した条件 1)のキャッシュ要件を満たす。次に、Cache-control によるキャッシュ可否を無視するため、CacheIgnoreCacheControl を設定する。CacheContol ヘッダの値が上記条件 4),5)に該当した場合でもキャッシュ可能とする。また、CacheIgnoreNoLastMod を設定し、上記条件 6)に該当した場合でもキャッシュ可能とする。加えて、CacheStoreNoStore と CacheStorePrivate を設定することで、条件 7),8)に該当した場合でもキャッシュに保持できるようにする。Expires ヘッダがないコンテンツに対して Expires を設定するため、HTTP のヘッダに先読みサーバで expires を設定する。この設定により条件 5), 6)に該当した場合でもキャッシュに保持できる。一方で Vary ヘッダをコンテンツのヘッダから削除することで、条件 9)を満たしキャッシュを保持可能にする。最後に、基本的にはキャッシュは URL で一意に決定されるが、Cookie や UserAgent によってキャッシュが変わる可能性がある。これを防ぐため CacheIgnoreHeaders を設定する。以上の設定によって、今回適用する乗換検索に関してはキャッシュが有効になる。

apache におけるコンテンツのキャッシュは、URL を利用して行う。まず、コンテンツの URL をハッシュ関数 md5 によって 128bit のハッシュ値とする。結果 32 文字の 16 進数が生成されるが、apache はこの 32 文字を 22 文字に圧縮する。この 22 文字を利用して、キャッシュを保持する。キャッシュが存在するかどうかはこの 22 文字の 16 進数を利用して

判定する。

また、apacheにおいてキャッシュされたコンテンツは、コンテンツの有効期限によって利用されなくすることは可能であるが、自動的に消去されない。したがって、キャッシュは定期的にシステムが明示的に削除する必要がある。この処理は、apacheに付属しているhtcachecleanと呼ばれるソフトウェアを利用して、定期的にキャッシュディレクトリ中の無効になったコンテンツを削除するようにした。

5.4. ユーザ入力事前送信

ここまで先読みサーバで先読みを行う際の手法を述べた。Pre_search_engineではユーザの入力の事前送信が必要となるため、その機能を実装する必要がある。本提案では、先読みサーバに入力の事前送信を行い、本要求を送信するためのページとしてポータルを用意した。本ポータルにおいて行う処理は、次の4つである。

- 1) 乗換検索要求の際の時刻情報を、現在時刻に設定
- 2) 検索タイプや利用交通機関を初期設定
- 3) ユーザの入力を先読みサーバに事前送信
- 4) 先読みサーバに対してコンテンツを本要求

ポータルはphpで記述し、先読みサーバに配置する。ユーザがアクセスを行った場合、その時の時刻をもとにポータルを自動生成し、ユーザに送信する。時刻や検索タイプ、利用交通機関の入力欄はセレクトボックスやラジオボタンで生成するため、これらの初期値も設定してユーザに送信する。



図 9 ポータル

ユーザの入力を先読みサーバに事前送信する際は、**Javascript** を利用して先読みサーバの **Pre_search_engine** に **POST** で送信する。事前送信するタイミングは、自由記述される乗車駅、降車駅に関しては、入力欄のフォーカスが外れた時点で送信を行う。この機能は、**javascript** の **onBlur** イベントを利用して実装した。従って、自由入力欄に関しては駅名が完全に入力された状態で先読みサーバに送信される。現在時刻や検索タイプなど、セレクトボックスやラジオボタンで値が設定されているものに関しては、値が変更された段階で送信するようにした。この機能は **javascript** の **onChange** イベントを利用して実装した。作成したポータルは図 9 に示すとおりである。

5.5. コンテンツ要求制御，データベース更新

データベースの中で、**UserSearch** はユーザの検索要求のうち **WebSiteRanking** に存在しない乗降駅データを保持する。したがって、**UserSearch** に適切なタイミングでデータを更新、挿入する必要がある。この処理を行うため、スクリプト **redirector** を用意した。このスクリプトは **php** で記述し、ユーザがポータルを利用して動的コンテンツの本要求を行ったとき、以下の処理を行う。

- ユーザの要求を実際のサービス提供者サーバへ転送する
- ユーザの求めた乗降駅の組がデータベースに存在しない場合追加する
- ユーザの求める乗降駅の組がデータベースに存在していた場合、検索されたことを示すデータベースのカラム **times** を 1 増加させる

5.3 節で述べたように、**apache** は **md5** によって **URL** のハッシュ値を求め、得られたハッシュ値を利用してキャッシュを保持する。またキャッシュの検索はこのハッシュ値をもとに行われる。したがって、先読みにおいてキャッシュに保持した動的コンテンツを参照するには、先読み時の **URL** と同一の **URL** をユーザも利用しなければならない。ここで同一の **URL** とは、**GET** によって送信される乗車駅、降車駅などのクエリの順序も一致している必要がある。この二つの **URL** を一致させるため、**redirector** を利用してユーザが要求する **URL** を先読み時の **URL** と一致させ、この **URL** にリダイレクトを行う。

また、本システムではデータベースに存在しない乗降駅の組に関しては、先読みの対象とならず先読みサーバにキャッシュを保持できない。先読み候補を増加させるため、データベースに存在しない乗降駅の組がユーザから検索された場合は、その組をデータベースに追加する。

6. システムの評価

前節で提案するシステムの実装を述べた。本節では実装したシステムを実際のサーバで動作させ、評価を行う。

6.1. システムの動作環境

先読みサーバは以下の表 11 に示す性能を持つサーバに対して実装した。

表 11 先読みサーバを実装したサーバ

OS	Ubuntu 14.04 LTS
CPU	Core 2 Quad Q9550 @ 2.83GHz
Memory	4GB
DiskSize	145GB
Ethernet	1Gbps

また、実際のサービス提供者サーバに対して先読みを無制限に実行することはできないため、実験を行うにあたってサービス提供者サーバとなるサーバを用意した。このサーバのシステム構成を以下の表 12 に示す。このサーバは、58KB のファイルをユーザの要求によって配信する。また、実装を行うに当たり以下の表 13 に示すアプリケーションを利用した。

表 12 評価用に用意したコンテンツ配信用サーバ

OS	Ubuntu 14.04 LTS
CPU	Core 2 Duo E6405 @ 2.13GHz
Memory	4 GB
DiskSize	160GB
Ethernet	1Gbps

表 13 利用したアプリケーション

機能	アプリケーション	備考
サーバ	Apache 2.4.10	
スクリプト	php 5.5.15	スレッドセーフで動作
データベース	MySQL 5.5.41	オンメモリで動作

システム評価を行う際は、実際にモバイル端末を大量に用意することは困難であるため、端末が要求を行う動作を模したクライアント動作用コンピュータを用意した。このコンピュータは、先読みサーバに対して同一のプライベート IP 空間に存在し、ネットワーク遅延などを制御する dummynet[40]を動作させているコンピュータを介して先読みサーバに接続する。また、クライアント動作用コンピュータとサーバは、1ms 以内の遅延時間で通信可能である（図 10）。

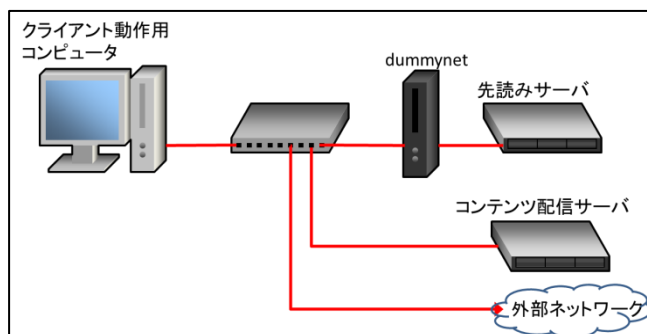


図 10 システム評価時ネットワーク

評価を行うに当たり乗換検索に利用する乗降駅の組が必要になる。このデータは、国土交通省が公表している大規模交通センサスの第 11 回調査[41]を利用した。この調査は、三大都市圏別に駅間の移動人員を集計し公表している。本評価実験では首都圏の駅間移動人員を利用した。

本システム評価では、以下の流れで作成したシステムの効果を評価する。

- 1) システム性能評価
- 2) 実際のサービス提供者サーバに対する応答性能評価
- 3) 長時間動作による応答性能評価
- 4) 先読みに関するヒット率

1) では同時にアクセスする端末数や、先読みサーバで生成するクエリの数を変化させ、処理にかかる時間を評価する。2) では、実際のサービス提供者サーバに対して処理を適用し、動的コンテンツを入手するまでの時間がどのようになるか、評価する。3) では、一定時間先読み処理を継続し、データ取得時間がどのように変化するか、またハードウェアのリソース使用はどの程度か評価する。4) では、先読みを行うに当たりどれだけヒットさせることができるか、また先読みによって入手した動的コンテンツは、問い合わせの可能性

のある乗降駅間の何パーセントをカバーできたか、先読みサーバにキャッシュを保持した動的コンテンツは複数回利用されたかを評価する。

6.2. システム性能評価

(1) 複数(10~200)のクライアントから同時に先読み要求を受けた場合、先読みサーバはサービス提供者サーバへ問い合わせするためのURLをどれだけの時間で生成可能かを評価した。ここで同時とは、先読みサーバに先読み要求が送信されるタイミングが全クライアントで1ms以内の差で一致している状態である。先読み時検索される駅数(最小1, 最大100)も変化させ、性能を測定する。測定は4回行い、4回の測定の平均を取った。また測定は連続に行うのではなく、一回の測定ごとに10秒程度のインターバルを置いて行った。

例えば10個のクライアントが先読み要求を行い、その結果すべての先読み要求に対して乗降駅の組が1つだけ存在した場合は、先読みによって生成されるURL数は10である。また、200個のクライアントが先読み要求をして、その結果すべての要求に対し乗降駅の組が100存在した場合、先読みによって生成されるURL数は20000である。この時の結果を図11に示す。

クライアント数に着目した場合、同時に先読み要求をするクライアントの数が10個であり、それぞれの先読み要求によって1個ずつURLが生成され、結果的に10個のURLを生成するにかかる時間は8msである。また、10個のクライアントからの先読み要求が結果

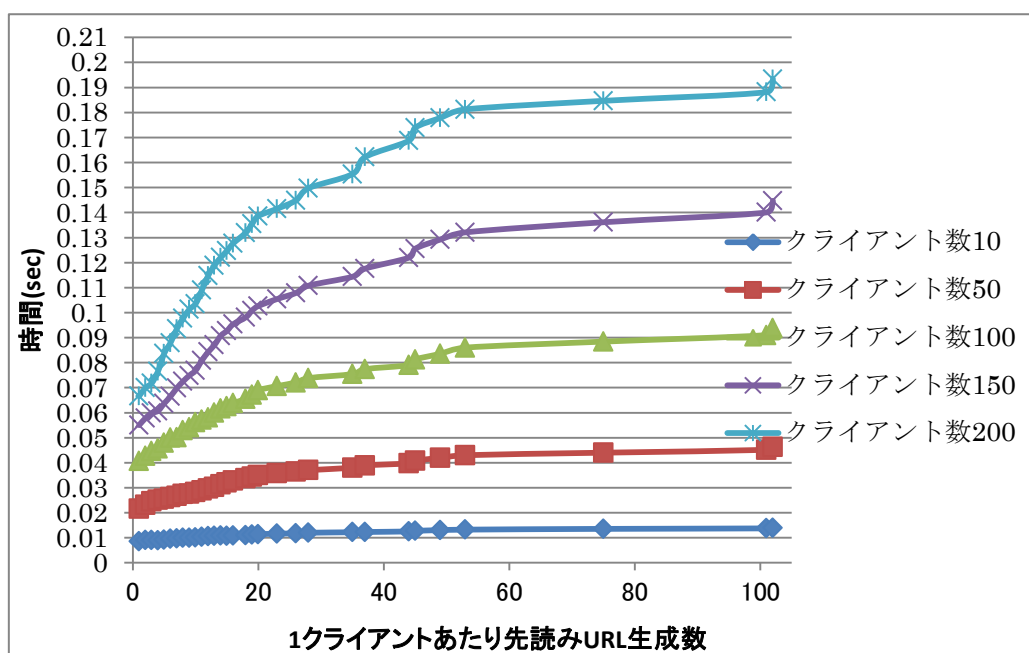


図 11 同時接続クライアント数別の URL 生成個数と生成時間の関係 (中央値)

として 1000 個の URL を生成するのにかかる時間は 13ms である。200 個のクライアントから要求があり、結果的に 20000 の URL を生成するのにかかる時間は 190ms である。いずれの場合も、ユーザがポータルを利用して降車駅を入力している間に、先読み用 URL は生成可能と考えられる。

先読み数に着目した場合、クライアント数と URL 生成にかかる時間の関係を示すグラフを図 12 に示す。先読み要求に対して 53 個の URL を生成する場合、200 クライアントでも URL は 180ms で生成可能である。

(2) 複数(10~300)の先読み用 URL に関して、先読みサーバが GET を行いコンテンツを

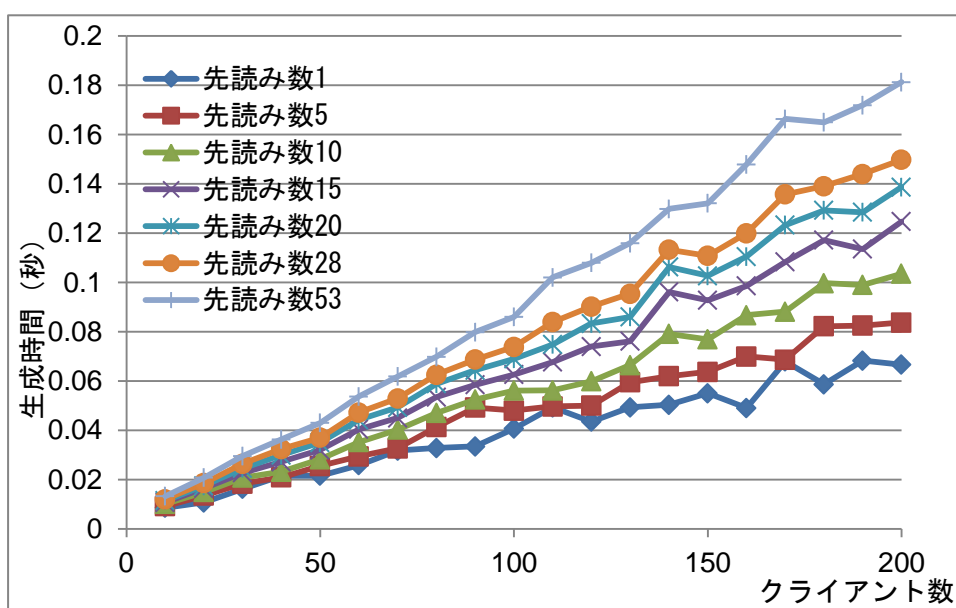


図 12 先読み数別の同時接続クライアント数と生成時間の関係 (中央値)

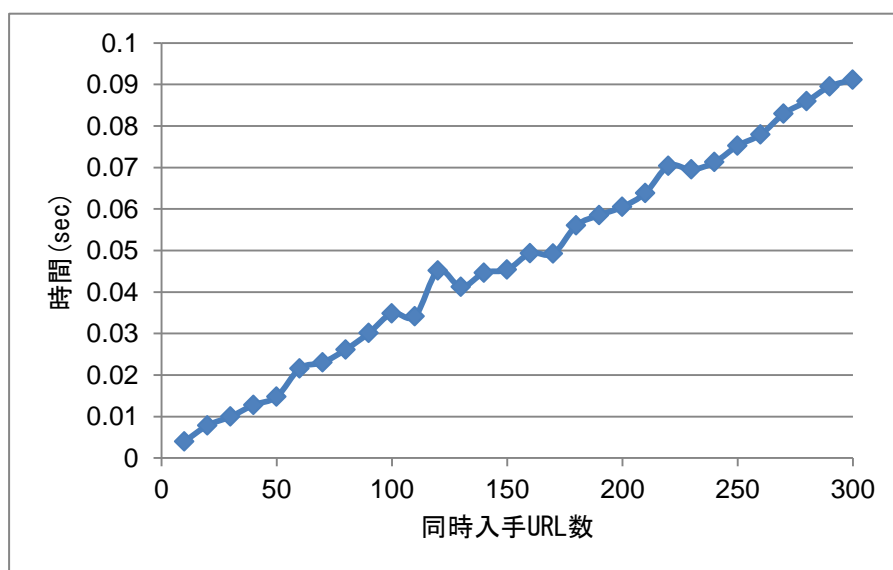


図 13 同時入手 URL 数とキャッシュするまでの時間の関係

入手して、apache のキャッシュに保持するまではどれだけの時間がかかるかを評価する。本評価においても、4 回評価を行いその平均を取った。問い合わせを行うサーバは評価用に用意したコンテンツ配信サーバに対して行った。ネットワーク遅延は 1ms 以内で、コンテンツ配信サーバからも 1ms 以内にコンテンツが送信される。

生成した URL を GET でコンテンツ配信サーバに送信し、キャッシュに格納されるまでの時間を図 13 に示す。

これより、10 個の URL からデータを入手しキャッシュするまでにかかる時間は 4ms であり、300 個の場合でも 90ms でキャッシュに保持している。URL 数とキャッシュに保持するまでの時間は線形に増加している。

6.3. 応答性能評価（実サービス提供者）

システム実装において、先読みサーバの機能を Yahoo 乗換検索、Jorudan 乗換検索の各サービスに対して実装した。ここでは、サービス提供者に対して実際に先読み処理を用いてコンテンツを取得すると、ユーザがコンテンツを入手する時間はどれだけ短くなるかを評価する。実サービス提供者に対して過度に問合せ要求を送信すると、DOS 攻撃となる可能性がある。そのため評価は 2 回行い、その平均を取った。また、問い合わせを行う際は連続的に行うのではなく、クライアント数が増加する毎に 10 秒ごとのインターバルを置いて問い合わせを行った。

Yahoo 乗換検索に対して、コンテンツ取得までの時間を測定した結果を図 14 に示す。縦

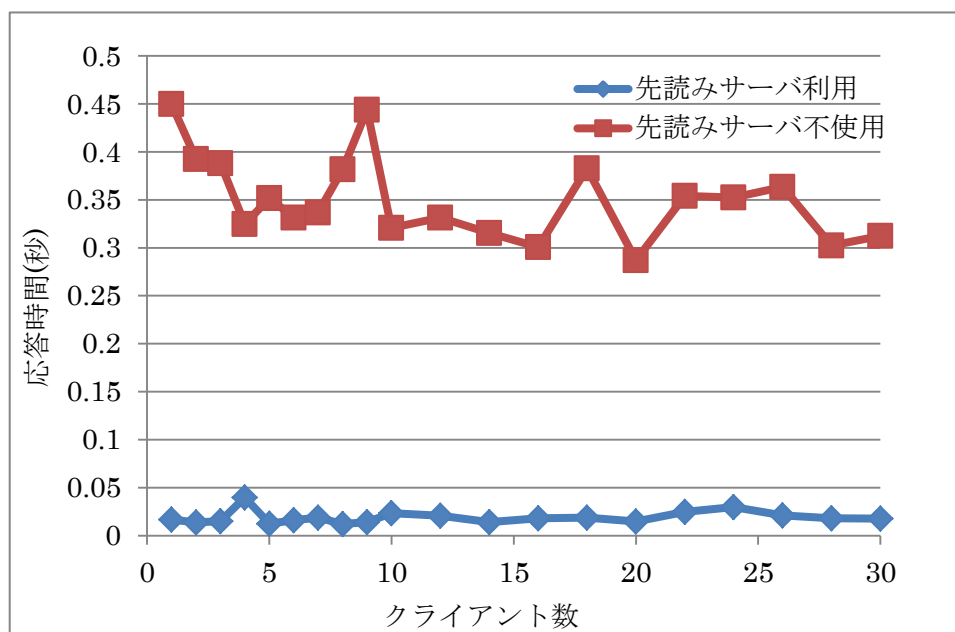


図 14 Yahoo 乗換検索に対する先読み有無による応答速度の違い

軸は応答時間，横軸は同時に要求を行ったクライアント数である．先読みサーバを利用しなかった場合は約 300ms から 450ms の応答時間となっていることと比較し，先読みサーバを利用した場合は約 15ms 程度の応答時間となっている．先読みサーバを利用していない場合の応答時間が安定していないのは，yahoo 乗換検索に関するサービス提供者サーバまでのネットワーク遅延や，サービス提供者サーバでの動的コンテンツ生成時間による影響と考えられる．一方先読みサーバを利用した場合，ネットワーク遅延が 1ms 以内の位置にコンテンツが存在しているため，サーバの処理時間分の応答遅延のみとなり，動的コンテンツの応答性が改善する．これより，先読みを行うことでユーザの要求よりも早く動的コンテンツを先読みサーバのキャッシュに保持できるため，ユーザからの本要求が来た場合は先読みサーバ内のキャッシュからコンテンツを提供できる．

Jorudan 乗換検索に対しても同様に評価を行った．その結果を図 15 に示す．Jorudan 乗換検索の場合も，先読みサーバを利用した場合はクライアント数が増加しても約 20ms でユーザにコンテンツを配信することが可能である．

以上の評価から，いずれの場合も先読みサーバを利用することで，動的コンテンツの応答性を向上させることが可能である．

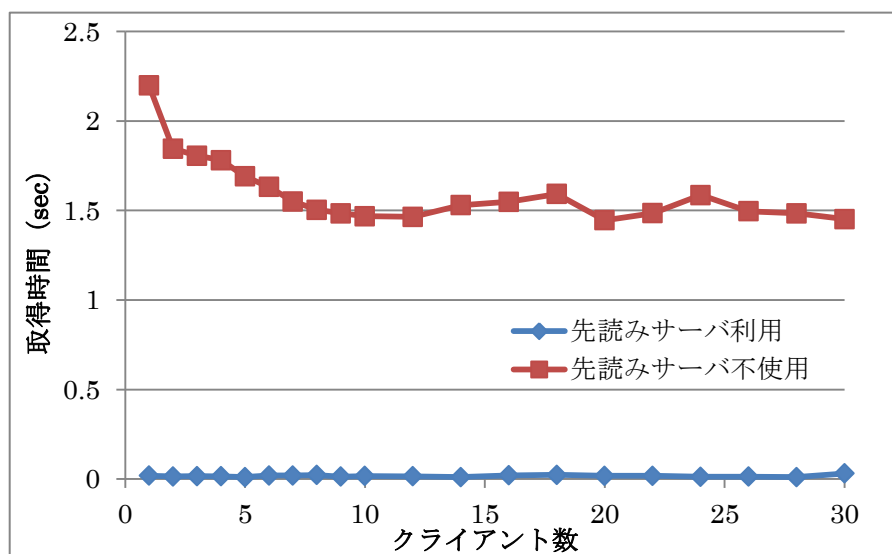


図 15 Jorudan 乗換検索に対する先読み有無による応答速度の違い

6.4. 応答性能評価（実験用コンテンツ配信サーバ）

6.3 節の評価において、実サービス提供者サーバに対して実験を行った結果、応答速度の改善が得られた。しかし 6.3 節では、散発的な問い合わせによって応答時間を評価していた。本節では連続的に先読みサーバに問い合わせを行い、連続的な問い合わせが動的コンテンツの応答性にどのような影響を与えるかを評価する。

本評価においては、1 秒ごとに先読みサーバに問い合わせる回数を変化させた。回数は 4 回/秒、6 回/秒、8 回/秒、10 回/秒とした。また、先読みを行う個数は、最大で 40 個となるようにした。連続問い合わせ継続時間は 1 時間とした。評価を行う際、実際のサービス提供者サーバに大量の問い合わせを行うことは、サービス提供者サーバに対しての攻撃となる可能性がある。そのため同一ローカルネットワーク内に配置したコンテンツ配信サーバに対して、問い合わせを行うようにした。コンテンツ配信サーバでは、実際のサービス提供者サーバに問い合わせた際の動作を模擬するため、要求が来た際は指定時間待機した後コンテンツを応答させるようにした。この時間は、Yahoo 乗換検索の応答時間平均値 307ms を用いた。

ここではまず、秒間問合せ回数の変化によって応答時間がどのように変化するかを評価する。次に、各秒間問合せ回数について、一分ごとに応答速度がどのように変化するかを示す。また、評価を行っている際のシステムリソース占有率についても述べる。

秒間問合せ回数による応答時間の変化とハードウェアリソース

秒間問合せ回数を変化させ、コンテンツの応答時間がどのように変化するかを評価した。その結果、応答時間は図 16 に示すとおりとなった。

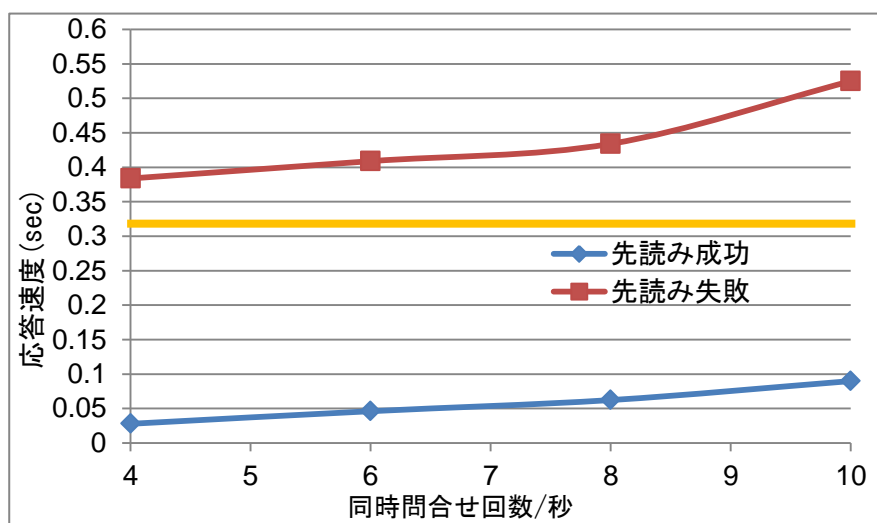


図 16 秒間問合せ回数と応答速度の関係

図中 0.3 秒のラインが、先読みサーバとコンテンツ配信間の遅延となる。先読みが成功した場合は、秒間問合せ回数が 10 回となっても、応答速度は約 90ms となっており、先読みサーバを用いない場合よりも応答速度は向上している。一方で、先読みに成功していない場合は、先読みサーバを用いない場合よりも最大で 200ms ほど処理に時間がかかっている。

また、図 16 は 1 時間の応答時間の平均であるが、1 時間の評価における 1 分ごとの応答時間の平均を図 17 に示す。秒間問合せ回数が少ない間は、応答時間は評価を通して安定しているが、問合せ回数が増加するにつれ応答時間のばらつきが大きくなっている。

また、1 時間連続して問い合わせを行った時のハードウェアリソースについても評価した。CPU 使用率を図 18 に。ディスク書き込み量を図 19 に示す。また、秒間問合せ回数別の

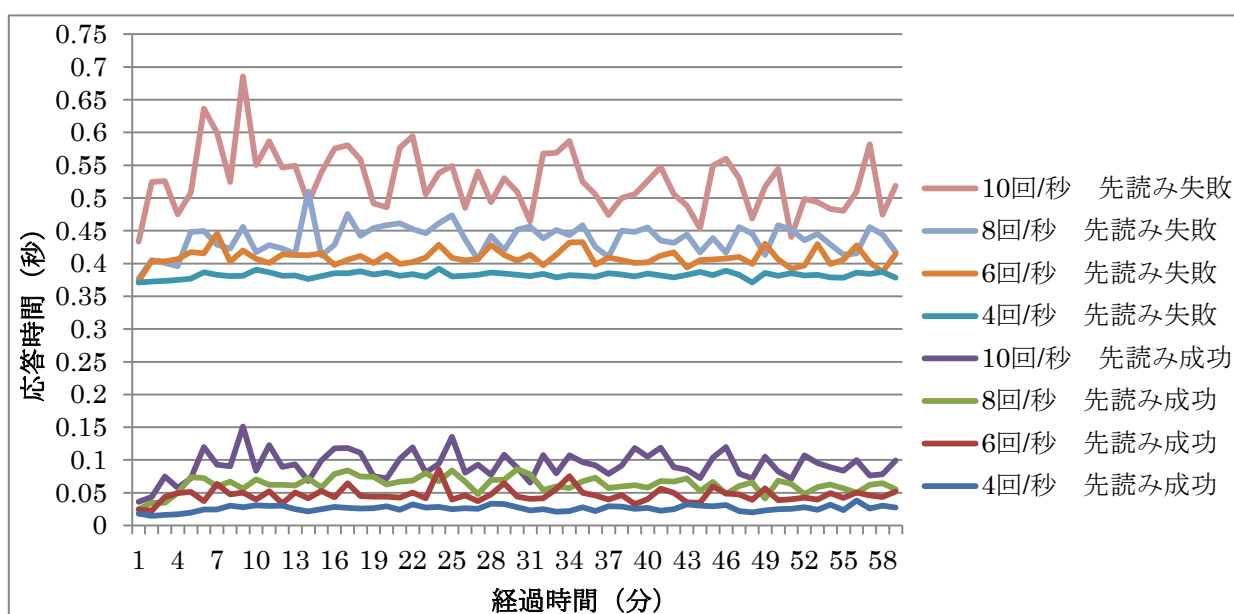


図 17 秒間問合せ回数別、1 時間の応答速度の変化

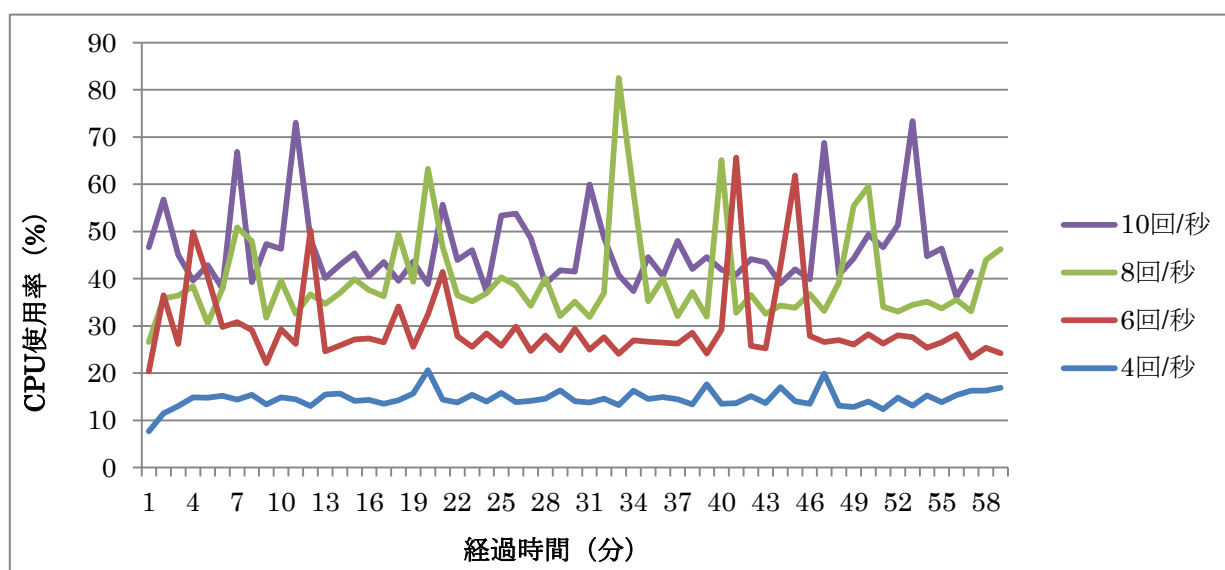


図 18 秒間問合せ回数別、1 時間の CPU 使用率の変化

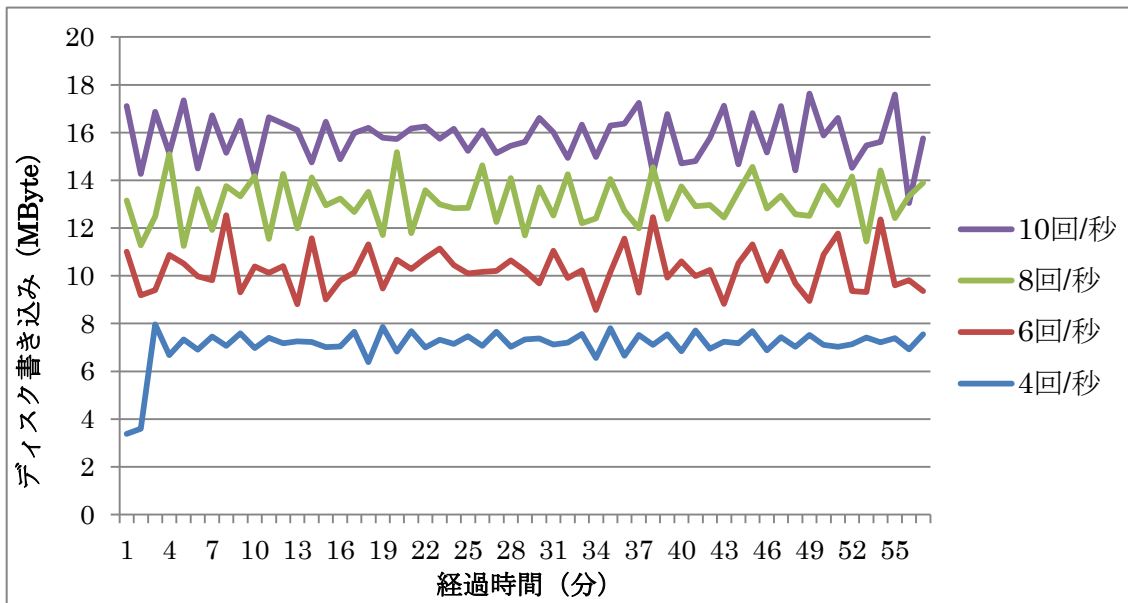


図 19 秒間問合せ回数別，ディスク書き込み量の変化

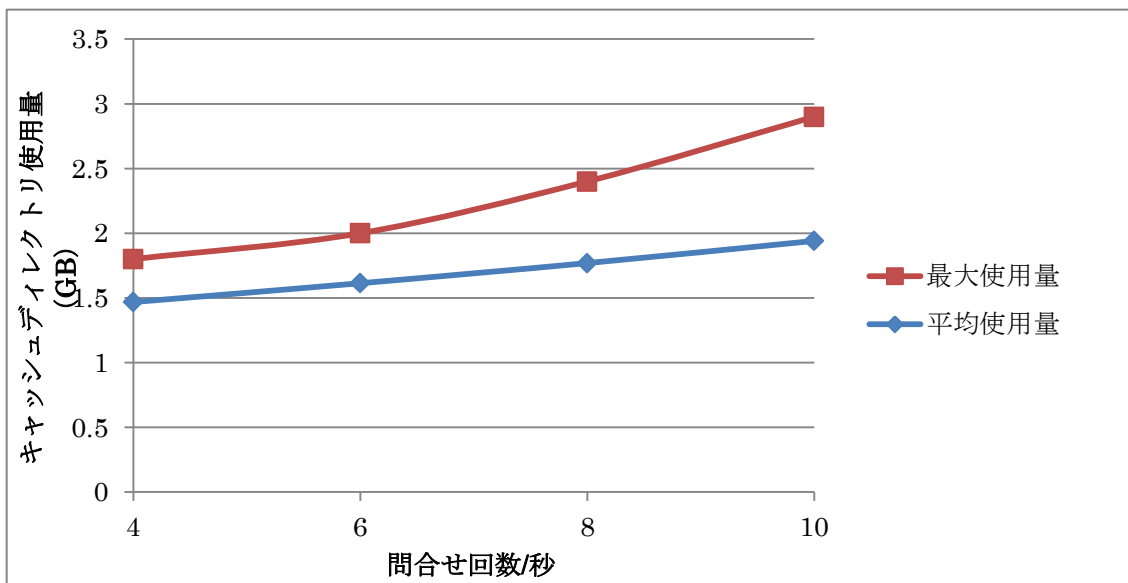


図 20 問合せ回数とキャッシュディレクトリ使用量の関係

評価におけるキャッシュを保持するディレクトリの使用量の平均値と最大値に関するグラフを図 20 に示す。

10 回/秒の場合で CPU 利用率は 40%前後となった。ディスク書き込みバイト数も秒間問合せ回数が増加するごとに増加している。これは、一時間毎のユーザからの問い合わせ回数が増加した結果、先読みの数も増加し、より多くの先読み結果がキャッシュに書き込まれるためである。また、キャッシュディレクトリの平均使用量と最大使用量は、問合せ回数が増加するごとに大きくなっている。評価を開始した時点でのディレクトリ使用量は同

一であるため、不要となったキャッシュの削除量に対して、先読みを行った結果のキャッシュ量が上回っていることがわかる。

6.5. ヒット率評価

これまで、提案したシステムによって乗換検索の動的コンテンツの応答性が、どのように改善するかを評価してきたが、本節では、先読み処理のヒット率に関して評価を行う。ここでは次に示す2つのヒット率の評価を行った。

- 1) ソーシャル情報を利用しない場合
- 2) ソーシャル情報を利用する場合

1) の場合は、先読み個数の違いによりユーザの要求する動的コンテンツをどれだけの確度で先読みサーバに用意できるかを評価した。また、先読みを利用せず、キャッシュ機能のみの場合にヒット率はどのようになるか評価した。さらに、ヒット率に加え、先読みサーバにおいてキャッシュの再利用を行うことにより、先読み回数がどのように変化するかについても評価した。

2) の場合は、1)の場合に加え、ソーシャル情報を利用した場合のヒット率の影響を評価した。

本評価では、先読みサーバに対してどのような順序で問合せが行われるかというシナリオを作成する必要がある。このシナリオは、文献[39]で示した駅間の移動人員と、ジョルダン株式会社の子会社であるコンパスティービー株式会社の公開している広告取扱媒体資料[42]に記載されている時間毎の問合せ数情報を基に作成した。

文献[42]には一日の各時間に何回程度 *Jorudan* 乗換検索に対して検索が行われているかの情報が概算で記載されている。一日のうち、各時間の検索回数は表 14 のとおりである。

一日の検索回数総計は 128 万 5000 回であった。一方で、文献[41]に存在する駅間移動人員の総計は、2291 万 42 人であった。次に駅間移動人員総計と一日の検索回数の比を求めた。これは、 $128 \text{ 万 } 5000 / 2291 \text{ 万 } 42$ となり、0.0561 となった。この値を各駅の移動人員数に乗ずることにより、各駅間が何回検索されるかを計算した。このため、駅間の移動人員が多いほど、検索回数は多くなる。

求めた各駅間の検索回数を超過しないよう、検索される駅間を 6 時から 25 時までランダムに割り振り、一日の検索されるシナリオを作成した。

表 14 Jorudan 乗換検索における時刻別検索回数

時刻	検索回数/時	検索回数/分	時刻	検索回数/時	検索回数/分
6時	30,000	500	16時	62,000	1,030
7時	50,000	830	17時	62,000	1,030
8時	62,000	1,030	18時	61,000	1,020
9時	61,000	1,020	19時	60,000	1,000
10時	62,000	1,030	20時	58,000	970
11時	62,000	1,030	21時	60,000	1,000
12時	62,000	1,030	22時	70,000	1,170
13時	60,000	1,000	23時	79,000	1,320
14時	58,000	970	24時	59,000	980
15時	59,000	980	25時	30,000	500

また、ユーザの検索した乗降駅の情報を保持するデータベース UserSearch は、評価開始直後はデータが空の状態からスタートさせるようにした。

1) ソーシャル情報を利用しない場合

Pre_search_engine を用いた場合、ユーザの求める動的コンテンツをどれほど先読みサーバに用意できるかを評価したグラフを図 21 に示す。Pre_search_engine での先読みを利用せず、先読みサーバのキャッシュ機能のみを用いた場合、ユーザの要求する動的コンテンツを先読みサーバで用意できる率は 6%から 10%前後であった。一方で先読みサーバを利用することで、ユーザの求める動的コンテンツを先読みサーバから提供できる可能性が上昇する。100 個先読みを行った場合は、約 85%のユーザに対して、先読みサーバからコンテンツを提供することが可能である。また、データベースに乗降駅の情報が追加されていくことで、先読みの精度が上昇していることがわかる。

また、キャッシュを行うことによって先読み回数はどのようになるのかを評価したグラフを図 22、図 23 に示す。このグラフは、評価に利用したシナリオの特定の 1 分間において、先読みがどれだけ行われたかを積算している。キャッシュ確認ありの場合は、すでにキャッシュに保持されているコンテンツに関しては先読み処理を行わないため、先読み回数を減らすことができる。一方キャッシュ確認なしの場合は、ユーザからの先読み要求毎にキャッシュに保持しているかにかかわらず先読みを行うため、先読み回数が線形に増加する。

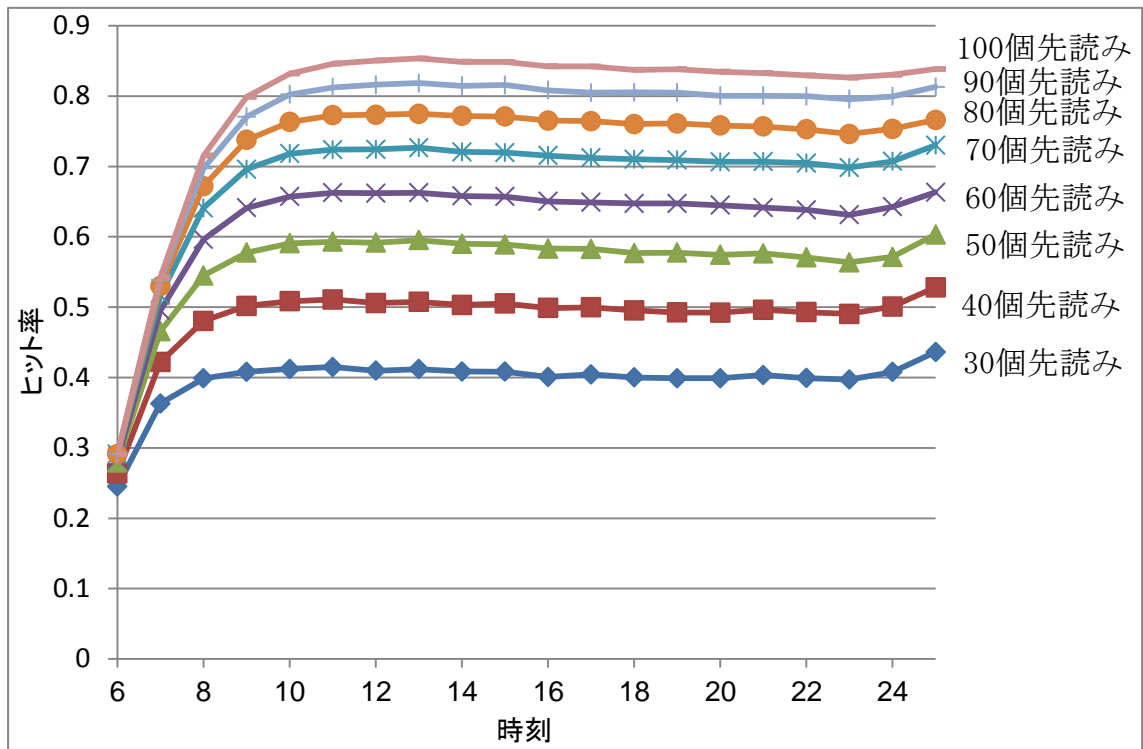


図 21 先読み個数別、シナリオに対する先読みヒット率

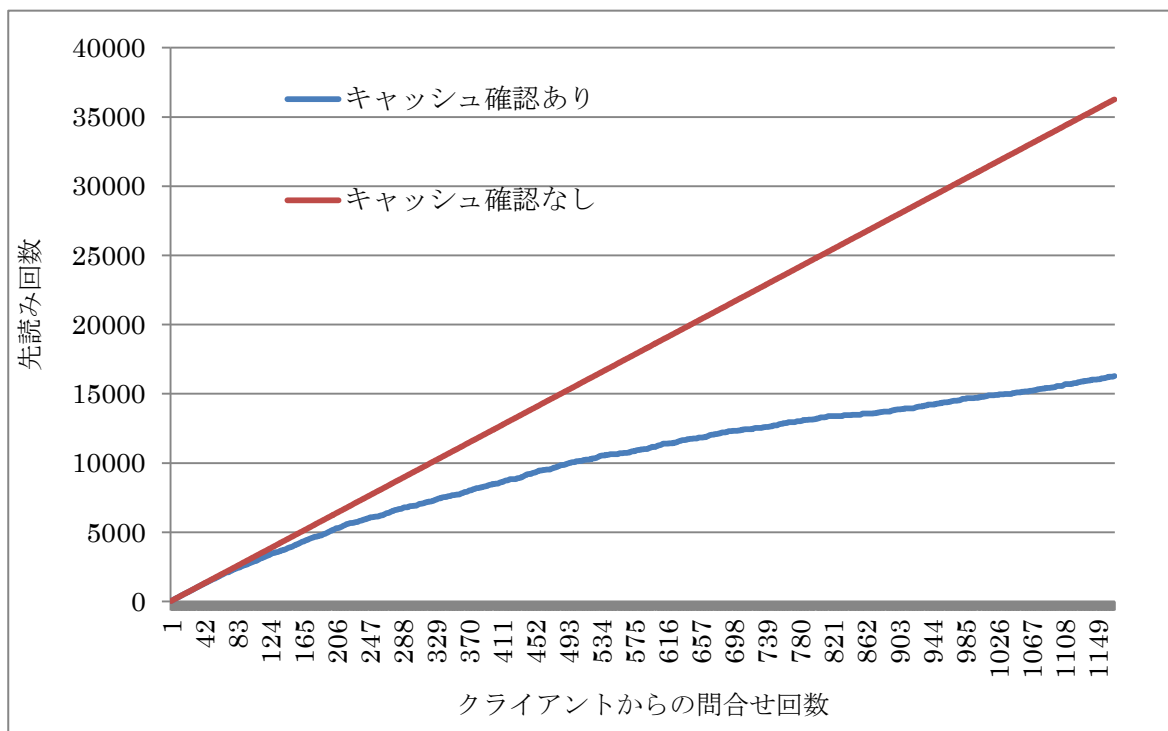


図 22 30 個先読み時、1 分間の先読み回数積算値

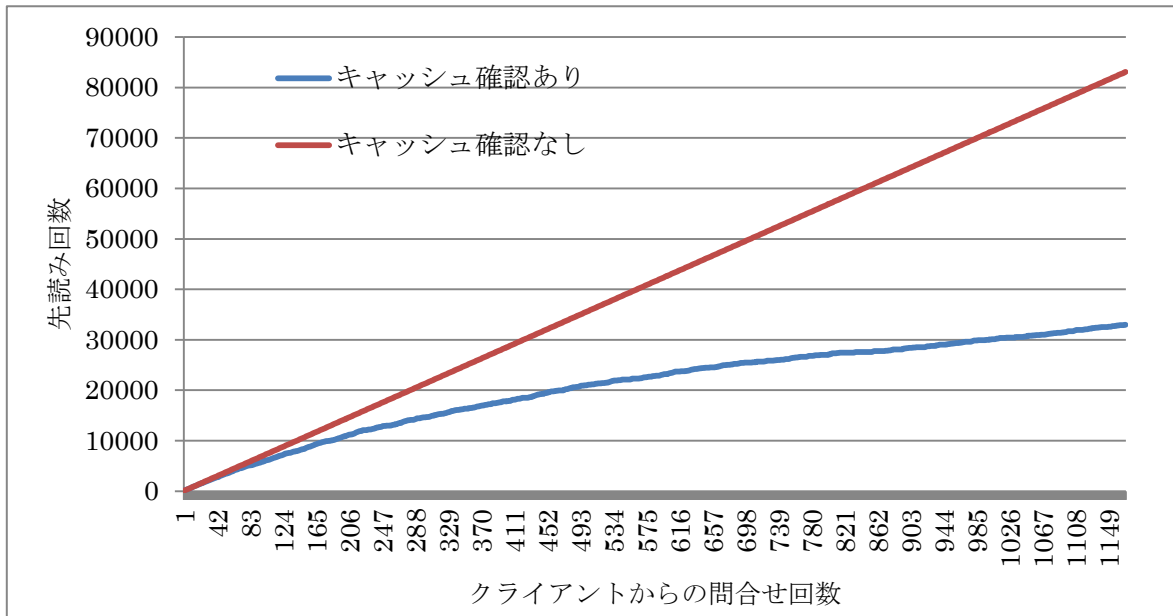


図 23 70 回先読み時、1 分間の先読み回数積算値

2) ソーシャル情報を利用した場合

ソーシャル情報を利用した場合についてヒット率を評価した。ソーシャル情報を利用するシナリオは、実際の列車遅延情報を利用して評価時シナリオを作成した。2014 年 12 月 13 日から 12 月 19 日にジョルダンライブに投稿された情報を取得し、その中で最も一日の投稿数が多かった 12 月 16 日と、最も投稿の少なかった 12 月 14 日のデータを利用した。その中で、ジョルダンライブへ長時間多くの書き込みが行われた 19 時から 23 時までの間を、評価に用いる書き込みとした。評価用シナリオは、得られたソーシャル情報から路線名を抽出し、路線に属する駅のうち乗車駅として選ばれる可能性の高い上位 5 駅を選択し、乗車駅とした。次に大都市交通センサスの駅間移動人員を利用して、選択した乗車駅を発駅とする移動人員の多い上位 20 組を選択した。最後に、その駅間ごと最小 1、最大 10 としてランダムに問合せ回数を設定し、遅延が発生した際の問合せシナリオとした。このシナリオと、ソーシャル情報を利用しない場合に利用したシナリオを混合し、最終的な評価用シナリオを作成した。

評価用シナリオを用いた先読みヒット率の時間変化のグラフを図 24 から図 27 に示す。図 24 はジョルダンライブへの書き込みの多い場合のシナリオに対して Social_crawler を利用しなかった場合、図 25 は Social_crawler を利用した場合である。また図 26 は書き込みの少ない場合のシナリオに対して Social_crawler を適用しなかった場合、図 27 は

Social_cralwer を利用した場合である。書き込み回数が多い場合でも少ない場合でも、Social_cralwer を使用した場合としない場合の差はほとんど得られなかった。

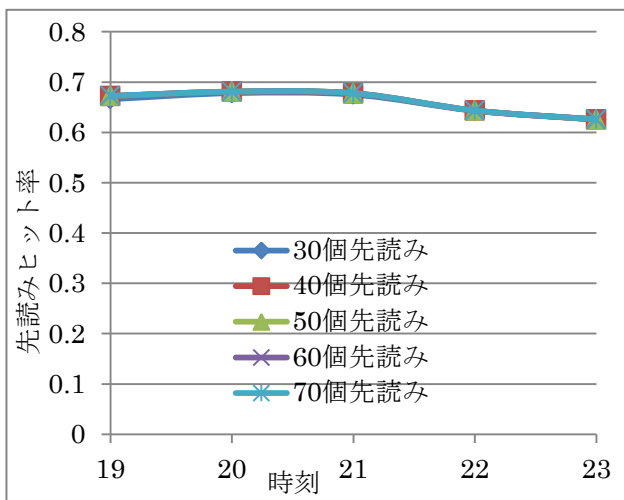


図 24 書き込み数最大 Social_crawler なし

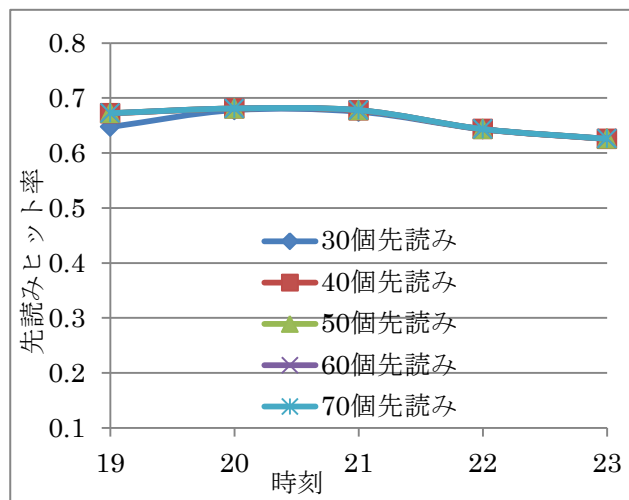


図 25 書き込み数最大 Social_crawler あり

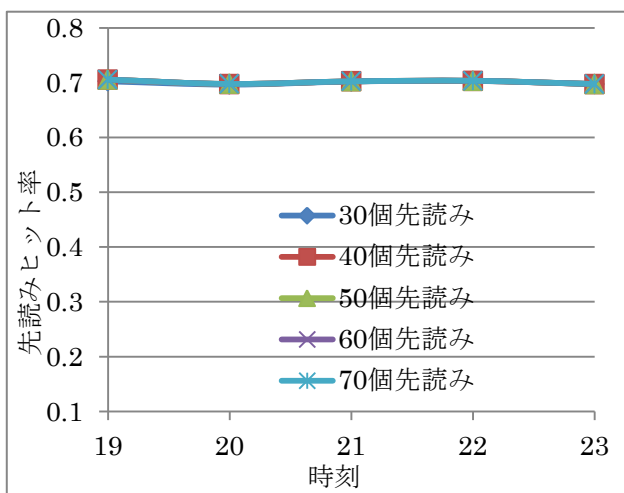


図 26 書き込み数最小 Social_crawler なし

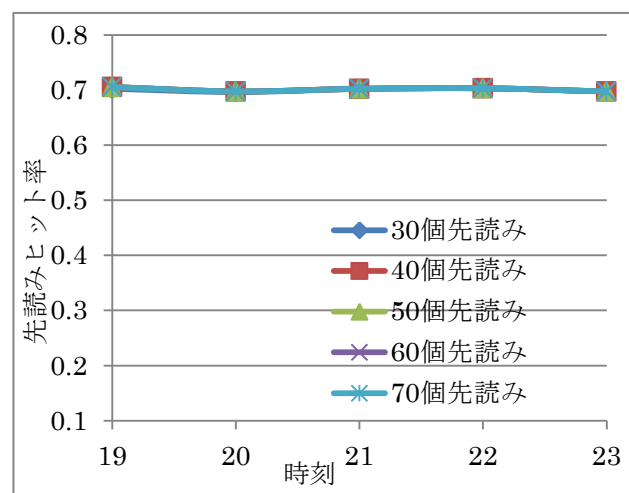


図 27 書き込み数最小 Social_crawler あり

7. 考察

前節で行ったシステムの評価を行った内容について考察を行う。また、今後の課題としてシステムをより改善するためにはどうすればよいか、より一般的な動的コンテンツに対して適用するにはどうすればよいかを考察する。

7.1. 評価内容に関する考察

システム性能評価に関する考察

評価においては、先読みクエリの生成と URL から先読みを行いキャッシュするまでの時間を測定した。この二つの時間を合計したものが、実際にユーザからの先読み要求に対して動的コンテンツを用意できる時間であると考えられる。評価に利用したサーバでは、300 個のコンテンツを要求してキャッシュに保持する処理に 90ms を要した (図 13)。300 個の URL を生成する処理は、10 個のクライアントが同時に先読み要求を出し、先読みサーバがそれぞれの先読み要求に対して 30 個の URL を生成したときの処理に等しい。このときの処理は約 12ms で終了している (図 12)。またキャッシュを用意する処理にかかる時間は、URL 数の増加に比例して伸びており、その処理時間の変化はほぼ線形である。従って、先読み処理にかかる時間に大きな影響を与えるのはキャッシュを用意する部分である。

先読み処理は、ユーザの検索したい動的コンテンツの検索語の一部が入力された時点で行われるため、先読み処理はユーザが検索語を入力し終わる前に終了していればよい。本評価の場合、先読み処理を行っている間にユーザは降車駅を入力する。300 個の動的コンテンツの先読み処理にかかる時間は約 100ms であり、ユーザが降車駅を入力するのにかかる時間よりも速く先読み処理が終了すると考えられる。

従って、本論文が提案するように、ユーザの入力の一部を事前入手することにより、ユーザの入力が終了する前に動的コンテンツを用意することができると考えられる。

応答性能評価（サービス提供者側サーバ）に関する考察

先読みサーバを用いた場合、ユーザから見たコンテンツの応答速度が向上するだけでなく、応答速度の分散が小さくなっていることがわかる。評価に用いたネットワークでは、クライアントと先読みサーバは 1ms 以内の遅延かつ 1Gbps の帯域で接続していた。このネットワークの特徴によって、応答時間の分散が小さくなっていると考えられる。

先読みサーバによる応答時間の短縮は、先読みサーバに保持されているキャッシュの影響による。先読み処理はユーザの要求するコンテンツをユーザに代行して入手しているだけであり、キャッシュに保持しなくてはコンテンツの応答性改善は得られない。コンテン

ツの応答性に寄与するのは、先読みサーバにおけるキャッシュ機能である。

応答性能評価（実験用コンテンツ配信サーバ）に関する考察

秒間問合せ回数が増える毎に応答時間が伸び、ユーザ応答時間の分散も大きくなっていることがわかる。ハードウェアリソースを考慮すると、CPU利用率には余裕があるが（図 18）ディスク書き込みバイト数は秒間 10 回の問合せ数で、ディスクに約 16MB/s 書き込んでいることがわかる（図 19）。評価においてキャッシュディレクトリは 1 台のハードディスク上に作成した。1 台のハードディスクにキャッシュの書き込みとキャッシュ探索のための読み込みが同時に発生しているため、秒間問合せ回数が増加するほど、キャッシュ処理がボトルネックとなっていることが考えられる。秒間問合せ回数が増加しても応答性を安定させるためには、ハードディスク上ではなくメモリ上などにキャッシュディレクトリを作成することが解決策として挙げられる。

また、秒間問合せ回数が増加するごとに、キャッシュディレクトリ使用量の平均値と最大値が増加している。乗換検索に関しては、コンテンツの有効期間は 1 分であり、頻繁にキャッシュディレクトリ中の期限切れキャッシュを消去している。しかしキャッシュディレクトリの使用量は増加しているため、キャッシュ消去が先読みによるキャッシュ追加に追いついていないことがわかる。評価では 1 時間の連続稼働に関して評価しているが、実運用上は先読みサーバを止めることはできない。従って、先読みしてもユーザに利用されないような無駄な先読みを減らすよう、先読み処理を行う必要がある。

ヒット率評価に関する考察

コンテンツの応答性が向上するのは、先読みサーバにユーザの要求するコンテンツがキャッシュされているためである。しかし、キャッシュ機能のみではそのコンテンツを最初に要求したユーザに対しては応答性の向上が得られない。ヒット率の評価から、先読み機能によって、100 個先読みをした場合は約 85%のユーザの要求の応答性を向上できることがわかる（図 21）。

一方で、先読み個数を増加させてもヒット率は線形に向上せず、ヒット率の変化が小さくなっている。例えば、30 個から 40 個へ先読み個数を増加させた場合は 10%ヒット率が上昇したのに対し、90 個から 100 個では改善は 4%程度であった。乗換検索に関しては、乗車駅として選択される可能性のある駅は 1405 駅であった。また乗車駅に対する降車駅の数 100 個を上回る駅は 254 駅存在した。これより降車駅数が 100 未満の駅が 1151 駅あり、この駅の全体からの割合が約 85%である（図 28）。また、降車駅の数が多いほど、先読みの成功率が低くなる。従って、ある検索語が先読み入力として先読みサーバに送信さ

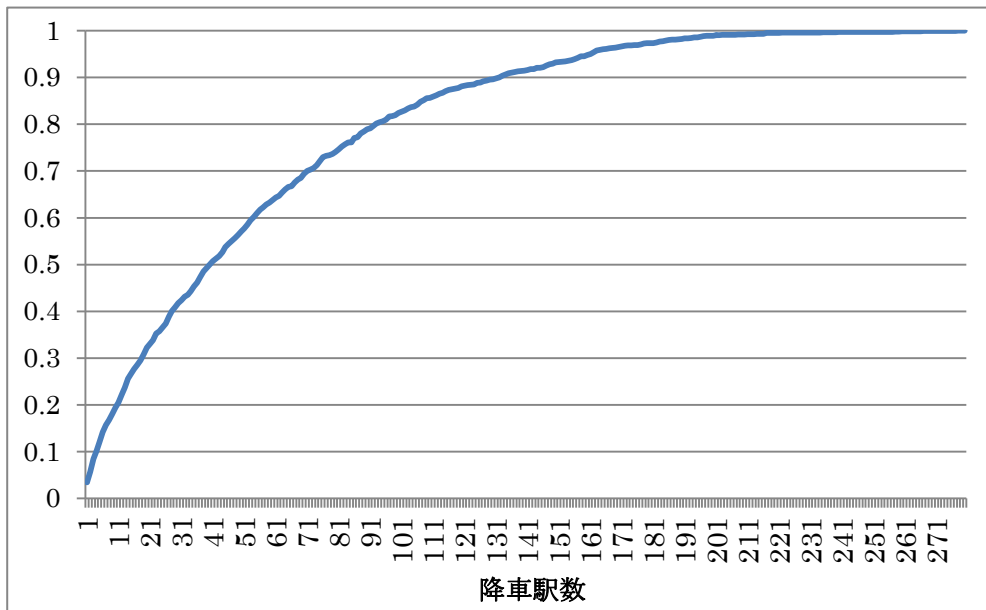


図 28 降車駅数に着目した乗車駅数の累積分布

れた際に、先読み候補となる語の数のばらつきが小さいほど、先読みサーバはより高精度でユーザが要求するコンテンツを用意できる。一方でばらつきが大きいほど全体的なヒット率は低下する。

1 分間の先読み回数の積算に関して、時間がたつほど先読み回数を減らすことができる (図 22, 図 23)。これはすでに先読みサーバのキャッシュに存在しているコンテンツに関しては先読みをしないためである。もし先読み処理をクライアント側で行った場合、他のユーザとのコンテンツの再利用が不可能なため、キャッシュ確認なしのグラフが示す回数ネットワーク内に先読み要求が発生する。先読みサーバをネットワーク内に配置することで、コンテンツの再利用ができていないことが確認できる。

しかし、先読み個数を多くすることで一度も利用されない動的コンテンツも同時にキャッシュに保持する問題がある。1 分間の検索回数が 1000 回の場合、ヒット率が 70% のとき最大でも 700 個のキャッシュしか利用されず、残りは一度も利用されないまま廃棄される。これより、より少ない先読み個数でヒット率を向上させる必要がある。

ソーシャル情報を用いた場合と用いていない場合での、ヒット率には有意な差が得られなかった (図 24 から図 27)。先読み処理による応答性向上は、最初にコンテンツを要求したユーザに対して恩恵がある。Pre_Search_engine はユーザの要求によって先読みを行い、Social_Crawler はソーシャル情報によって先読みを行う。しかしソーシャルメディアに書き込まれる情報は、多くのユーザがそれに関する情報で検索する可能性が高い。従って、ソーシャル情報を利用した先読みは、ユーザからの初回先読み要求を代行しただけであり、

ヒット率の点ではソーシャル情報を利用しなかったものと有意に差はなかったものと考えられる。

ソーシャル情報も利用して作成したシナリオにおいては、1回の問い合わせに対する先読み回数が少ない場合でもヒット率が高くなっている。これはソーシャル情報を利用したことにより同一の要求が複数のユーザから送信される回数が増加したためである。これにより先読みヒット率が少ない先読み回数でも上昇していると考えられる。

7.2. 今後の課題

7.1節の議論で、先読みサーバで先読みを行うことで、これまで難しかった最初にコンテンツ要求を行うユーザに対しても応答性向上が可能になり、中間サーバでキャッシュを行うことでより多くのユーザに対して応答性を改善できることを述べた。一方で今回の評価において、ソーシャル情報は有用な先読み情報とはならなかった。また、利用されない無駄な先読みを減らすため、より少ない先読み個数でヒット率の改善を行う必要があることも分かった。本節ではそれらに関して検討する(1)。また、より一般的な動的コンテンツに対して適用に関して考察する(2)。

(1) ソーシャル情報の活用・効率の良い先読み手法の検討

ソーシャル情報はある時点で一回のみ取得されるのではなく、一定期間連続して書き込まれるものと考えられる。例として、12月16日21時から23時にかけてジョルダンライブに投稿された内容のうち、ある路線に関する書き込みがあった時間を表15に示す。

表 15 各路線に関する 21 時から 23 時までの書きこみ時刻

埼京線	21 時 39,41,42,49,52,55,59 分 22 時 2,6,8,11,14,16,18,21,26 分
山手線	21 時 44,47,49,52 分
京浜東北線	21 時 53 分 22 時 11,13,21,23,28,36,38,40,43,45,55,58,59 分

表 15 に示す通り、ソーシャル情報として書き込まれる内容は単発的ではなくある時間内に継続して行われるという特徴があると考えられる。従って、ユーザの要求によって行われる先読みと比較し、ソーシャル情報を用いる先読みはより長い時間検索される可能性が高いと考えられる。本評価においてはユーザの要求による先読み、およびソーシャル情報に

よる先読みの有効期間はともに 1 分に設定している。しかし、ソーシャル情報を利用してキャッシュとして保持したコンテンツの有効期間を長くすることで、無駄な先読みが発生しないようにすることができると考えられる。また、ユーザの先読みは過去の検索回数が多い順で行っている。ソーシャル情報を利用する際は、検索回数が多い順ではカバーできない検索可能性の低い駅間について重みを付けて先読みを行えば、全体的な先読みヒット率を向上させることが可能と考えられる。このような検索アルゴリズム考えることが今後の課題である

本論文におけるシステム実装では、先読みが終了する前にユーザからの本要求が来た場合の処理を考慮していない。先読みサーバからコンテンツをユーザに送信するためには、ユーザからの本要求が来る前にキャッシュにコンテンツを保持している必要がある。しかしコンテンツがキャッシュに存在していない場合は本要求もサービス提供者サーバに到達してしまうため、先読みが無駄となってしまう。これを防ぐために、先読みを行っているコンテンツに関してはユーザの要求を先読みサーバで一時的に止める処理を行う必要がある。この処理を実装し妥当性を評価することが今後の課題と考えられる。

(2)一般的な動的コンテンツへの適用に関する考察

最後に、本論文では動的コンテンツを乗換検索に限定して実装を行い評価した。実際には乗換検索以外の動的コンテンツも存在する。それらについて、本システムがどのように適用可能か考察し、今後の課題を検討する。

本システムにおいて適用した乗換検索は、自由記述欄が複数ある動的コンテンツであった。このため、ある一方の検索語を利用して別の検索語を補完し、先読み要求を行うことができた。複数の検索語を用いて生成される動的コンテンツにおいては、本システムが提案した先読み手法と同様を用いることができると考えられる。しかし乗換検索は検索語が駅名に限定されているが、他の動的コンテンツに関しては限定されていない。この場合、語の共起を利用して、検索される可能性のある候補を絞る手法が考えられる。語の共起とはある単語が出現した場合に他の限られた単語が出現することである。ある単語 A が出現した場合に他の単語 B が頻出するとき、A と B は共起度が高いとなる。この語の共起度を持つことで、ある単語が検索語として入力された場合に別の検索語を候補として絞ることができると考えられる。

また、複数の検索語ではなく単一の検索語のみで動的コンテンツが生成されることが考えられる。この場合、検索語がすべて入力されたのちに先読みを行っていたら、ユーザの

本要求までにコンテンツを用意できないと考えられる。このような動的コンテンツに対しては検索語が確定する前に入力された文字を先読みサーバが入手して、ユーザの入力が終了する前に先読みを行う必要がある。しかしユーザの入力した先頭の文字だけで先読みを行うことは、精度の問題で難しいと考えられる。この場合、先読みサーバで先読みを行う前に、1文字目だけでなく2文字目以降の情報も適宜入手し、先読み候補を絞っていく処理が必要である。

8. 結論

本論文では、近年多くの人に利用されているがその特徴から応答性を向上させることが難しい動的コンテンツに関し、「動的コンテンツ先読み機構を持つネットワーク内キャッシュシステム」を提案した。提案したシステムを、動的コンテンツとして需要のある Yahoo 乗換検索, Jorudan 乗換検索に関して実装し、システムの有効性を評価した。その結果、先読みサーバを用いることにより、両コンテンツの応答性を最短 10ms, 先読みサーバの負荷が高い場合でも 100ms 以内にユーザに送信することができ、コンテンツのユーザから見た応答性を向上させた。また、最大で 85%のユーザに対してコンテンツの応答性を向上させることができた。加えて先読みサーバを複数のユーザが利用することにより、先読みサーバを用いない場合より先読み回数を削減できることができた。

システム評価から得られたデータによって、動的コンテンツはメモリ上にキャッシュとして保持する必要があるなど、コンテンツ応答性をより向上させるための知見を得た、先読みを行う入力として、ユーザからの要求だけでなくソーシャル情報を用いる場合は、どのような状況であればソーシャル情報は有効な入力となるかを検討した。

最後に、乗換検索だけでなくより一般的な動的コンテンツに対して本システムを適用する場合には、語の共起を利用して検索語の先読みを行う必要があるなど、考慮しなければならないことについての知見を得た。

参考文献

- [1] Google map, “<https://www.google.co.jp/maps?hl=ja>” (2015年1月10日アクセス)
- [2] ジョルダン乗換検索, “<http://www.jorudan.co.jp/>” (2015年1月10日アクセス)
- [3] 食べログ, “<http://tabelog.com/>” (2015年1月10日アクセス)
- [4] ジョルダン株式会社, 2014年9月期決算事業内容・展開について, “http://www.jorudan.co.jp/ir/data/ir/accounts/20141114_meeting.pdf” (2015年1月10日アクセス)
- [5] 財団法人インターネット協会, インターネット白書 2012, インプレスジャパン, 2012年, pp.46
- [6] Eytan Adar et,al. “The Web Changes Everything: Understanding the Dynamic Web Content.,” In Proc of the 2nd ACM International Conference on Web Search and Data Mining, pp.282-291, 2009
- [7] Linden G Geeking with Greg “<http://glinden.blogspot.com/2006/11/marissa-mayer-at-web-20.html>”, (2015年1月10日アクセス)
- [8] Nielsen J, “Usability Engineering Morgan Kaufmann,” San Francisco,1994年 pp.135
- [9] Kyunghan. Lee, et, al. "Mobile data offloading: How much can WiFi deliver?," IEEE/ACM Transaction on Networking, Vol. 21, No. 2, pp. 536-550, 2013
- [10] Dae Sun Kim. et, al. "Efficient ANDSF-assisted Wi-Fi control for mobile data offloading," 9th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 343-348, Jul. 2013
- [11] Soumen Mitra, et,al. “Deployment Considerations for Mobile Data Offloading in LTE-Femtocell Networks,” In Proc of International Conference on Signal Processing and Communications, pp.1-6, 2014
- [12] 中嶋 信生ほか, 携帯電話はなぜつながるのか 第2版 知っておきたいモバイル音声&データ通信の基礎知識, 日経 BP 社, 2014年,
- [13] アプリケーションキャッシュの利用 -MDN, “https://developer.mozilla.org/ja/docs/Web/HTML/Using_the_application_cache”, (2015年1月10日アクセス)
- [14] squid : Optimising Web Delivery, “<http://www.squid-cache.org/>”,(2015年1月10日アクセス)
- [15] Jianliang Xu, “Caching and prefetching for Web content distribution”, Computing in Science and Engineering, vol.6, issue.4, pp.54-59, 2004

- [16] Srikanth Sundaresan, et, al. “Accelerating Last-Mile Web Performance with Popularity-Based Prefetching”, In Proc of the ACM SIGCOMM, pp.303-304, 2012
- [17] Tsozen Yeh, et, al. “Improving the Performance of the Web Proxy Server through Group Prefetching”, In Proc of the 6th International Conference on Ubiquitous Information Management and Communication, Article No.81, 2012
- [18] Wei-Guang Teng, et,al. ” Integrating Web Caching and Web Prefetching in Client-Side Proxies”, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, Vol.16, No.5, 2005
- [19] B. de la Ossa, et,al. “Improving Web Prefetching by Making Predictions at Prefetch”, in Next generation Internet Networks, 3rd EuroNGI Conference, pp.21-27, 2007
- [20] 庄野篤司, et, al. “フィンガープリントキャッシュを用いた動的 Web コンテンツの効率的な配信方式”, 情報処理学会論文誌 Vol.46, No.2, 2005
- [21] Wenzhong Chen, et,al. “Caching Dynamic Content on the Web”, Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on, pp.947-950, 2003
- [22] Junchang Ma, et, al. “TeCaS: A Template Caching System for Dynamic Web Pages”, In Proc of AICT-ICIW’06, pp.189-194, 2006
- [23] Lingli Zhang, “Model for Efficient Delivery of Dynamic Web Pages with Automatic Detection of Shared Fragments”, In Proc of 22nd Wireless and Optical Communication Conference, pp.475-480,2013
- [24] Yeim-Kuan Chang, et, al. “Caching Personalized and Database-related Dynamic Web Pages”, In International Workshop on Networking Architecture and Storages, pp149-153, 2006
- [25] 柴田 和祈, et, al. “動的 web アプリケーションのためのプリフェッチ機構”, 情報処理学会論文誌, Vol.54, No.2, pp.647-658, 2013
- [26] Dimitrios Lymberopoulos, et, al. “Pocket Web: Instant Web Browsing for Mobile Devices”, In Proc of the 17th International Conference on Architectural Support for programming Languages and Operating Systems, pp.1-12, 2012
- [27] Andi Ahmad Dahlan, et,al. “Implementation of Asynchronous Predictive Fetch to Improve the Performance of Ajax-Enabled Web Applications”, In Proc of 10th International Conference on Information Integration and Web-based Applications & Services, pp.345-350,2008
- [28] James R. Challenger, et,al. “Efficiently Serving Dynamic Data at Highly Accessed Web Sites”, IEEE/ACM Transactions on networking, Vol.12, No.2, 2004

- [29] 米村 淳, 小花 貞夫, ネットワーク内での動的コンテンツ先読みに関する一考察, 2014年電子情報通信学会総合大会, 2014
- [30] 駅データ.jp,” www.ekidata.jp/”, (2015年1月10日アクセス)
- [31] 電車が止まったらまずソーシャル、そして始まった車両 IoT ,
“<http://itpro.nikkeibp.co.jp/article/Watcher/20140502/554583/>”, (2015年1月10日アクセス)
- [32] ジョルダンライブ, “<http://www.jorudan.co.jp/unk/live.html>” ,
(2015年1月10日アクセス)
- [33] Atsushi Yonemura, Teruyuki Hasegawa, Sadao Obana, “Proposal of In-Network Caching System with Dynamic Content Pre-fetching Mechanism,” International Conference of Mobile Computing and Ubiquitous Networking, 2015
- [34] TwitterAPI, “<https://dev.twitter.com/>”
- [35] Twitter REST API, “<https://dev.twitter.com/rest/public>”
- [36] Mecab Japanese Morphological Analyzer, “<https://code.google.com/p/mecab/>”
- [37] Apache mod_cache, “http://httpd.apache.org/docs/current/en/mod/mod_cache.html”,
(2015年1月10日アクセス)
- [38] RFC2616, “<https://www.ietf.org/rfc/rfc2616.txt>”, (2015年1月10日アクセス)
- [39] Caching Guide – Apache HTTP Server Version 2.4,
“<http://httpd.apache.org/docs/2.4/caching.html>”
- [40] Luigi Rizzo et, al. “Dummysnet: a simple approach to the evaluation of network protocols”, ACM Computer Communication Review, Vol.27, Issue.1, pp31-41, 1997
- [41] 国土交通省 大都市交通センサス 第11回調査 ,
“http://www.mlit.go.jp/sogoseisaku/transport/sosei_transport_tk_000007.html”,
(2015年1月10日アクセス)
- [42] コンパスティービー株式会社 広告取扱媒体資料 ,
“http://www.compasstv.net/business/ad_dl.html”, (2015年1月10日アクセス)

謝辞

本論文を進めるに当たり，多くの方からご指導，ご支援を賜りましたことを心より感謝申し上げます。

指導教員としてご指導いただきました小花貞夫教授に深く感謝申し上げます。博士前期課程2年間を通じて研究，並びに修士論文執筆において様々なご指導ご助言を頂きました。

副指導教員である寺田実准教授には修士論文執筆における助言を賜りましたこと深く感謝申し上げます。

湯素華助教には，研究の様々な点に置きまして的確な助言を頂きましたこと，深く感謝申し上げます。

また，研究機材の提供から研究における的確な指導，修士論文執筆における助言など，様々な指導を頂きました KDDI 研究所長谷川輝之グループリーダーに深く感謝します。

日々の輪講などにおいて様々なアドバイス，助言をしていただき，時には心を支えていただいた桂健太氏，齋藤淑氏，三井貴之氏，吉川潤氏をはじめとする研究室の皆様にも深く感謝いたします。

最後に，私を精神的，経済的に支えてくださった家族にこころより感謝いたします

付録 A 発表時資料

以下に学会発表における予稿，論文を掲載する．発表を行った学会，題目，及び掲載ページなどは以下に示す通りである．

- [A] 米村 淳, 小花 貞夫, ネットワーク内での動的コンテンツ先読みに関する一考察, 2014年電子情報通信学会総合大会, B-7-62, 2014
- [B] Atsushi Yonemura, Teruyuki Hasegawa, Sadao Obana, “Proposal of In-Network Caching System with Dynamic Content Pre-fetching Mechanism,” International Conference of Mobile Computing and Ubiquitous Networking, pp.100-101, 2015

ネットワーク内での動的コンテンツ先読みに関する一考察

A Study on Prefetching Dynamic Content using In-Network Cache

米村 淳 小花 貞夫
Atsushi Yonemura Sadao Obana

電気通信大学大学院情報理工学研究所
Graduate School of Informatics and Engineering, The University of Electro-Communications.

1. はじめに

近年, google map での乗り換え検索や食ベログなど, ユーザの入力によって表示結果の変わる動的コンテンツが増加している. ここでは, ユーザの検索要求 (以下, 要求) が web サーバに送られる毎に動的コンテンツが生成され, ユーザに送り返されるため, 一般的に静的コンテンツと比較して応答性が悪化すると考えられる. 本稿では, ネットワーク内での動的コンテンツの先読みによる応答性の向上を提案しその実現に向けた課題を示す.

2. 動的コンテンツ

2.1 動的コンテンツとその応答性

ここでは, 動的コンテンツとは, ユーザの入力した要求によって表示結果が変化するウェブページを指す. 静的コンテンツの場合, 一度コンテンツの要求があった時点でキャッシュサーバにキャッシュされれば, 以降はキャッシュサーバとクライアント間で通信を行うことができる. 一方動的コンテンツは, コンテンツの生成をコンテンツ提供者の web サーバ上で行うため, 要求ごとに web サーバとクライアントの間で通信が発生し, コンテンツ提供者の web サーバのみならず, ネットワークそのものの負荷も増大し, コンテンツの応答性も低下する.

2.2 動的コンテンツの先読み

前述した動的コンテンツを, キャリアや ISP 側のネットワーク内で先読みしてキャッシュとして保持しておくことで, 複数のクライアントからの同一の要求が発生した場合は, キャッシュからコンテンツを取得できるためネットワーク全体のトラヒックの軽減が可能となる. また通信は, 先読みにより要求が web サーバまで到達せず途中のキャッシュサーバまで済むため, ユーザから見たコンテンツの応答性向上が期待できる.

先読みの例を, google map での乗り換え検索を具体例として挙げる. ある路線が何らかの理由で運転見合わせとなった場合, 多くの人々が代替路線を検索すると考えられる. 先読みしない場合, すべての検索がサーバへ送信されサーバとネットワークの負荷が増加する. 先読みを行った場合, 検索要求が発生する前に結果の候補を用意できる可能性があるため, その場合は先読みなしと比較して応答速度が向上すると考えられる.

3. 関連研究

動的コンテンツのキャッシュ手法については従来から研究が行われており[1], 商用のシステム[2]も存在する. こ

れらはコンテンツ提供者側のサーバにキャッシュを設置する. この場合はコンテンツ提供者側までの通信は必ず発生するため, トラヒック削減効果は薄く, 応答性の向上も期待できないと考えられる. また, 携帯端末上で動的コンテンツをプリフェッチする研究が行われている[3]. この場合, 応答性は向上すると考えられるが, 同一のコンテンツを要求したい複数のユーザが存在した場合でも, それぞれが web サーバへ要求を行うためトラヒックの削減は期待できないと考えられる.

4. 実現方法の検討とその課題

先読み実現のため, 次の方法を用いることが考えられる.

- (1) ユーザの入力する検索要求の途中送信
- (2) 先読み用サーバでの結果候補の探索および web サーバへの要求代行

3節の乗り換え検索を例として方法を述べる. (1) まずユーザが出発地を入力した時点で, その入力先読み用サーバに送られる. (2) 送られた入力によってキャッシュを探索し, 存在しない場合は先読み用サーバが web サーバへの要求を代行する.

(1) を実現する際には, 要求の途中送信分のオーバーヘッドが発生するため, 如何にオーバーヘッドを抑えるかも課題となる. また, (2) においては, 先読み用サーバでの先読み検索代行と, 先読み結果のキャッシュ方法, 及びキャッシュの探索手法の効率の実現が課題であると考えられる.

5. おわりに

近年増大している動的コンテンツは, コンテンツ提供者のサーバ負荷とトラヒックを圧迫し, ユーザから見たコンテンツの応答性を悪化させる可能性があるという問題点を指摘し, それに対するネットワーク上での先読みという実現方法の検討とその課題を述べた. 今後は前述の課題を解決するための手法の考案, 検証を行う予定である.

謝辞

本稿執筆に多くのアドバイスを頂いた株式会社 KDDI 研究所の長谷川輝之グループリーダーに深く感謝します.

参考文献

- [1] K. Selçuk Candan *et al.* "Enabling dynamic content caching for database-driven web sites", Proc. of the ACM SIGMOD international conference on Management of Data, pp532-543, 2001
- [2] [http://technet.microsoft.com/ja-jp/library/dd239248\(v=ws.10\).aspx](http://technet.microsoft.com/ja-jp/library/dd239248(v=ws.10).aspx) (2014年1月7日アクセス)
- [3] Dimitrios Lymberopoulos *et al.* "PocketWeb : Instant Web Browsing for Mobile Devices", In Proc. of the 17th international conference on Architectural Support for Programming Languages and Operating Systems , pp1-12, 2012

Proposal of In-Network Caching System with Dynamic Content Pre-fetching Mechanism

Atsushi Yonemura¹, Teruyuki Hasegawa², Sadao Obana¹

¹Graduate School of Informatics and Engineering
The University of Electro-Communications
1-5-1 Chofugaoka, Chofu, Tokyo, Japan
a.yonemura@uec.ac.jp, obana@cs.uec.ac.jp

²KDDI R&D Laboratories, Inc.
2-1-15 Ohara, Fujimino, Saitama, Japan
teru@kddilabs.jp

Abstract—Recently, dynamic content, whose construction is changed dynamically according to an user's request, has been wide-spreading. Since many kinds of dynamic content are served from the origin servers, it is more difficult in general to improve a responsiveness of dynamic content than static one that can be cached at a location close to users. This paper proposes to cache such dynamic content to improve its responsiveness, and describes the design and implementation of our proposed in-network cache system which can pre-fetch and cache dynamic content near users.

Keywords— *Dynamic content; Pre-fetching; Caching;*

I. INTRODUCTION

Dynamic content (e.g. Google Transit Search [1], Tabelog [2]) is referred to as web pages whose construction, such as displayed images and processing results is varied in response to users' requests. It is generated at the origin web servers located in the service providers and is then transferred to the users. Static content, whose construction is fixed, can be cached at cache servers located near users and be served locally when the same request comes from other users. As for dynamic content which usually is not cached however, user may experience poor responsiveness.

Dynamic content is generated from the following elements;

- 1) *User specific information(e.g. cookie)*
- 2) *User input query*

Dynamic content generated from 1), or from 1) and 2) is called *user specific content*. Although the pre-fetching is effective in terms of the responsiveness, there is no benefit of caching due to little reuse probability. On the other hand, the one generated by only 2) depends on queries in users' requests which may be identical among multiple users, and caching such dynamic content is promising to give improved performance of web services. Based on the above brief insight, this paper proposes an in-network approach which can cache and also pre-fetch the latter type of dynamic content to improve a responsiveness of services and describes its system design and implementation.

II. RELATED WORK

There have been several approaches at either edge of a network. Caching dynamic content in cache servers located in each service provider has been already studied [3] and realized as commercial systems [4]. However, a user must communicate with servers inside a service provider to get the content. Thus,

the expected performance gain is limited because it can just shorten the time to generate dynamic content at origin servers inside the service provider.

On the other hand, a method of pre-fetching dynamic content on each client, i.e., a mobile terminal, has been studied recently [5]. By using this method, the response time could be shortened if the same user accesses the same content repeatedly. However, other users who want to get the same content still have to communicate with origin servers to get the content.

III. PROPOSED METHOD AND IMPLEMENTATION

Although some approaches exist for caching and/or pre-fetching dynamic content at ether edge of a network, their expected performance gain is limited. In order to improve performance, we propose a method to introduce facility of a special cache system with dynamic content pre-fetching into a telecom carrier's network.

A. In-network Caching with Dynamic Content Prefetching

As shown in Fig. 1, we implemented in-network cache and pre-fetching system named "Pre-fetching Server" nearby users. To improve the responsiveness and reuse the same dynamic content by multiple users, this system provides the following functions.

- 1) *Caching function*: The server stores both static and some types of dynamic content which has been received from origin servers and directly serves users' requests instead of forwarding the requests to the origin servers.
- 2) *Pre-fetching function*: The server, on behalf of users, fetches *in advance* dynamic content which users might want to get, later.

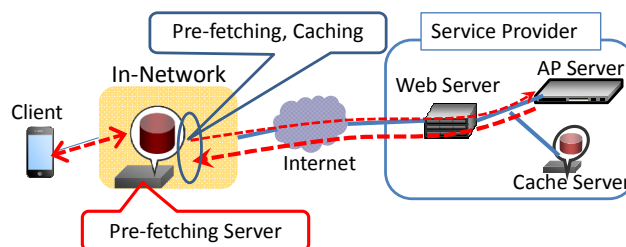


Figure 1 Pre-fetching Server and its deployment

For making the latter function effective, the Pre-fetching Server infers users' requests. For example in the case that the request is a search query, a part of the user input is sent to the Pre-fetching Server before the user clicks his submit button. Then, the server predicts complete queries based on some ranking analysis of messages stored in some social media such as Twitter, and Pre-fetches content from the origin server on behalf of the user.

It is difficult to implement a general Pre-fetching function covering all types of queries because their inquiry format has to be generated with conforming to each service. Therefore as a first step, we focus on applying the pre-fetching function to train transit search services, i.e., Yahoo transit search [6], Jorudan [7] and Navitime [8] for the following reasons. First, the input form variation of these systems is very limited, e.g. names of departure and destination stations. Second, train transit search is quite popular, especially in Japan. Moreover, many people often write short messages about train accidents into social media in a real-time manner.

B. Implementaion

To realize the above two functions, the Pre-fetching Server is implemented as a forward proxy using apache module "mod_proxy". The Caching function is implemented by using apache module "mod_cache". The Pre-fetching function is implemented by two scripts written in php and the database is managed by MySQL. We prepared a portal page (Fig. 2) as an input user interface of the Pre-fetching Server that can get the partial search query before the user clicks his submit button. We call this query "Pre-Send query".

Details of these two php scripts are as follows.

- **Pre Search Engine:** Pre-fetch dynamic content when a Pre-Send query arrives from a user.
- **Social Crawler:** Pre-fetch periodically (e.g. every 1minute) based on the analysis of tweets on Twitter.

Database has the following tables;

- **Ranking:** Search ranking published by service provider.
- **History:** Past queries arrived at the Pre-fetching Server.
- **Line/Station:** Association between rail line and stations.

These scripts pre-fetch content along the following steps:

[Pre Search Engine]

- 1-a) Receives a Pre-Send query from a user and extract search words.
- 1-b) Creates inquiry queries that are probably inputted by the user using Ranking and History tables.
- 1-c) Sends the queries created in 1-b) to the service provider.
- 1-d) Caches the content in search response.

[Social Crawler]

- 2-a) Collects tweets from Twitter every one minute.
- 2-b) Extracts information about train suspension from the collected tweets.
- 2-c) Gets station names that belong to suspended rail line from *Line/Station* tables and completes queries using *Ranking* and *History* tables.
- 2-d) Sends the queries created in 2-c) to the service provider.
- 2-e) Caches the content in search response.

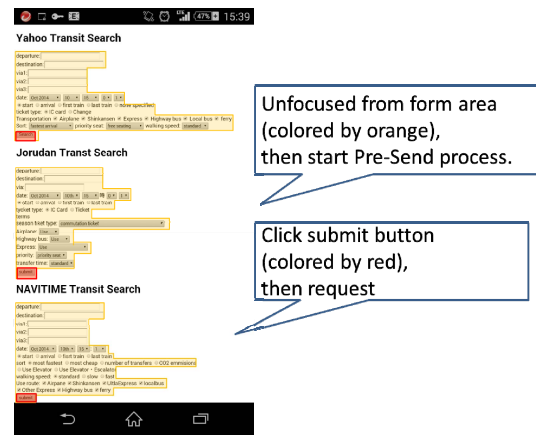


Figure 2 Display image and operation of portal

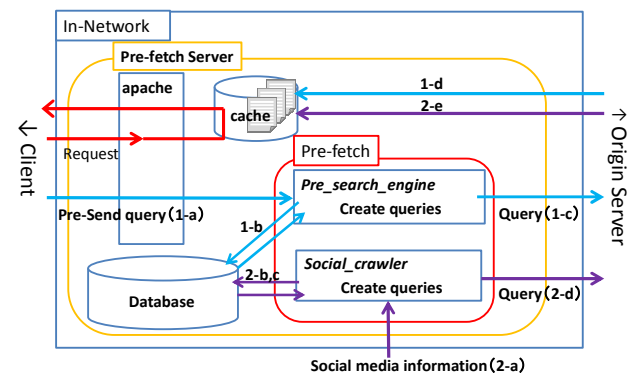


Figure 3 Entire data flow around Pre-fetching Server

Figure 3 shows the entire data flows around the Pre-fetching Server. When a user clicks submit button on the portal page, the user's request is sent to the Pre-fetching Server. If the pre-fetching is successful, the corresponding dynamic content has already existed and can be returned directly, and the responsiveness is improved. On the other hand, if the pre-fetching failed, dynamic content is served by the origin servers.

IV. CONCLUSION AND FUTURE WORKS

To improve a responsiveness of dynamic content, we proposed in-network cache system with dynamic content pre-fetching. Through an implementation of a proto-type system of transit search service, we confirmed the effectiveness and the feasibility of the proposed system. We plan to proceed the next step to realize a general pre-fetching mechanism.

REFERENCE

- [1] <http://maps.google.com/transit>
- [2] <http://tabelog.com>
- [3] K. Selçuk Candan *et al.* "Enabling dynamic content caching for database-driven web sites," Proc. of the ACM SIGMOD international conference on Management of Data, pp532-543, 2001
- [4] [http://technet.microsoft.com/ja-jp/library/dd239248\(v=ws.10\).aspx](http://technet.microsoft.com/ja-jp/library/dd239248(v=ws.10).aspx)
- [5] Dimitrios Lymberopoulos *et al.* "PocketWeb : Instant Web Browsing for Mobile Devices," Proc. of the 17th international conference on Architectural Support for Programming Languages and Operating Systems , pp1-12, 2012
- [6] <http://transit.loco.yahoo.co.jp/>
- [7] <http://www.jorudan.co.jp/>
- [8] <http://www.navitime.co.jp>

付録 B ソースコード

以下に先読みサーバに関するソースコードを示す。

pre_search_engine.php

```
1 <?php
2 #post 入手。DB 問い合わせは from to のみ。文字列整形のため抽出
3 $from = $_POST["from"];
4 $to = $_POST["to"];
5 $host = $_POST["host"]; #1:yahoo 2:jorudan 3:navitime
6 $hostname = "";
7 $i = 0;
8 #スペース改行などあれば削除
9 $from_trimmed = trim($from);
10 $to_trimmed = trim($to);
11 $from = urldecode($from_trimmed);
12 $to = urldecode($to_trimmed);
13 $wsr_id = array();
14 $us_id = array();
15 $str = "";
16 $time = date("Y-m-d H:i:s",time() - 60);
17 if(empty($from)){ #from がない
18     #なにもしない
19 }else if(empty($to)){ #from のみ存在
20     #sql へつなぐ
21     $dsn = 'mysql:host=localhost;dbname=transit;charset=utf8';
22     $usr = 'yonemura';
23     $pass = 'ekripssoft';
24     $dbh = new PDO($dsn,$usr,$pass);
25     try{
26         //$dbh = new PDO($dsn,$usr,$pass);
27
28         $sql_wsr = "select id, dst from WebSiteRanking_memory where src = \"\" .
29 $from . \"\" and last_searched < \"\" . $time . \"\" order by times desc,
30 last_searched desc limit 20;";
31         $sql_us = "select id, dst from UserSearch_memory where src = \"\" . $from .
32 \"\" and last_searched < \"\" . $time . \"\" order by times desc, last_searched desc
33 limit 20;";
34         /*get "to"*/
35         $i = 0;
36         foreach($dbh->query($sql_wsr) as $row){
37             $dsts[$i] = $row['dst'];
38             $wsr_id[] = $row['id'];
39             $i++;
40         }
41         foreach($dbh->query($sql_us) as $row){
42             $dsts[$i] = $row['dst'];
43             $us_id[] = $row['id'];
44             $i++;
45         }
46     }
47 }
```

```

41     }
42     //echo($i);
43
44     #問い合わせクエリ生成
45     #到着地以外のクエリをサービスごとに生成
46     }catch(PDOException $e){
47         print("PDO error: " . $e->getMessage());
48     }
49
50     if($host == 1){
51         $hostname = "transit.loco.yahoo.co.jp";
52         $before = "http://transit.loco.yahoo.co.jp/search/result?from=" .
urlencode($from) . "&to=";
53         $after = "&via=" . urlencode($_POST["via1"]) . "&via=" .
urlencode($_POST["via2"]) . "&via=" . urlencode($_POST["via3"]) . "&ym=" .
$_POST["ym"] . "&d=" . $_POST["d"] . "&hh=" . $_POST["hh"] . "&m1=" .
$_POST["m1"] . "&m2=" . $_POST["m2"] . "&type=" . $_POST["type"] . "&ticket=" .
$_POST["ticket"] . "&a1=" . $_POST["a1"] . "&shin=" . $_POST["shin"] . "&ex=" .
$_POST["ex"] . "&hb=" . $_POST["hb"] . "&lb=" . $_POST["lb"] . "&sr=" .
$_POST["sr"] . "&s=" . $_POST["s"] . "&expkind=" . $_POST["expkind"] . "&ws=" .
$_POST["ws"] ;//. "&kw=" . urlencode($_POST["kw"]);
54     }else if($host == 2){
55         $hostname = "www.jorudan.co.jp";
56         $before = "http://www.jorudan.co.jp/norikae/cgi/nori.cgi?eki1=" .
urlencode($from) . "&eki2=";
57         $after = "&eki3=" . urlencode($_POST["via"]) . "&via_on=1&Dym=" .
$_POST["Dym"] . "&Ddd=" . $_POST["Ddd"] . "&Dhh=" . $_POST["Dhh"] . "&Dmn1=" .
$_POST["Dmn1"] . "&Dmn2=" . $_POST["Dmn2"] . "&Cway=" . $_POST["Cway"] .
"&Cfp=" . $_POST["Cfp"] . "&C7=" . $_POST["C7"] . "&C2=" . $_POST["C2"] . "&C3=" .
$_POST["C3"] . "&C1=" . $_POST["C1"] . "&C4=" . $_POST["C4"] . "&C6=" .
$_POST["C6"]
"&S=%E6%A4%9C%E7%B4%A2&map1=&rfr=nr&pg=0&eok1=&eok2=&eok3=&Csg=1";
58     }else if($host == 3){
59         $hostname = "www.navitime.co.jp";
60         $before =
"http://www.navitime.co.jp/transfer/search?orvStationName="
urlencode($from) . "&dnvStationName=";
61         $after = "&month=" . urlencode($_POST["month"]) . "&day=" .
$_POST["day"] . "&hour=" . $_POST["hour"] . "&minute=" . $_POST["minute"] .
"&sort=" . $_POST["sort"] . "&wspeed=" . $_POST["wspeed"] . "&airplane=" .
$_POST["airplane"] . "&sprexprs=" . $_POST["sprexprs"] . "&utrexprs=" .
$_POST["utrexprs"] . "&mtrplbus=" . $_POST["mtrplbus"] . "&othexprs=" .
$_POST["othexprs"] . "&intercitybus=" . $_POST["intercitybus"] . "&ferry=" .
$_POST["ferry"] .
"&orvStationCode=&dnvStationCode=&basis="
$_POST["basis"] . "&ctl=020010&atr=2&detailset=1&init=";
62     }
63
64     $length = $i;
65     //if($length > 10){$length=10;}
66     #おのおの to に対して get 問い合わせクエリ生成
67     #QUERY として持つ必要あるのか?
68     for($j = 0; $j < $length; $j++){
69         $QUERY[$j] = $before . urlencode($dsts[$j]) . $after;
70         if($host == 1){
71             $QUERY[$j] .= "&kw=" . urlencode($dsts[$j]);

```

```

72     }
73     echo($QUERY[$j] . "\n");
74 }
75
76 $threads = array();
77 foreach($QUERY as $getquery){
78     $threads[] = new GetData($getquery,$hostname);
79     //echo($getquery);
80 }
81
82 $num = count($threads);
83 for($j = 0;$j < $num; $j++){
84     $threads[$j]->start();
85 }
86
87 try{
88     $update_wsr = "update WebSiteRanking_memory set last_searched = now()
where id = ?;";
89     $update_us = "update UserSearch_memory set last_searched = now() where
id = ?;";
90
91     $stmt = $dbh->prepare($update_wsr);
92     foreach($wsr_id as $elm){
93         $stmt->execute(array($elm));
94     }
95     $stmt = $dbh->prepare($update_us);
96     foreach($us_id as $elm){
97         $stmt->execute(array($elm));
98     }
99 }catch(PDOException $e){
100     print($e->getMessage());
101 }
102
103 foreach($threads as $th){
104     $th->join();
105 }
106
107 }else{ #ともに空白でなかったら
108
109 }
110
111 class GetData extends Thread{
112     private $host;
113     private $query;
114     function __construct($q,$h){
115         $this->host = $h;
116         $this->query = $q;
117     }
118

```

```

119     public function run(){
120
121         $header = array(
122             "Content-Type: application/x-www-form-urlencoded",
123             "Accept: /*/*",
124             "User-Agent: Mozilla/5.0",
125         );
126         $options = array(
127             "http" => array(
128                 "method" => "GET",
129                 "header" => implode("\r\n",$header),
130                 "proxy" => "tcp://localhost:8080",
131                 "request_fulluri" => true,
132             ),
133         );
134
135         file_get_contents($this->query,false,stream_context_create($options));
136         echo("url pre_searched\n");
137     }
138 }
?>

```

Social_cralwer.php

```

1 <?php
2 require_once'twitteroauth/twitteroauth.php';
3
4 $dsn = 'mysql:host=localhost;dbname=transit;charset=utf8';
5 $usr = 'yonemura';
6 $pass = 'ekripssoft';
7
8 define('CONSUMER_KEY', 'EKiR19m3lUqRcsn56zdJhz1y4');
9 define('CONSUMER_KEY_SECRET',
10 'v3q0IDVVRlJU9azHf1dVw6SPn8UFp2aUu50RhQxQ3zKrrVQbr3');
11 define('ACCESS_TOKEN', '2730992712-Is0hkQw9RMrndi4asgh261q5VLnPYk6XLf4eFQC');
12 define('ACCESS_TOKEN_SECRET', 'uFc7ZIzXdX55ckMpm57uoPg70Wyomdaj7wyBscqdv13bY');
13
14 $twitterOAuth = new TwitterOAuth(
15     CONSUMER_KEY,
16     CONSUMER_KEY_SECRET,
17     ACCESS_TOKEN,
18     ACCESS_TOKEN_SECRET
19 );
20 $mecab = new Mecab_Tagger();#mecab
21 $line = array(); #路線名配列
22 $station = array(); #駅名配列
23 $id = array();
24 $src = array();

```

```

25 $dsts = array();
26
27 $keywords = "ジョルダンライブ+-RT";#キーワード選択が重要なので要検証
28
29 $opt = array('q' => $keywords,
30   'lang' => 'ja',
31   'count' => '20',
32   'locale' => 'ja',
33   'result_type' => 'recent');
34
35 $json = $twitterOAuth->OAuthRequest(
36   "https://api.twitter.com/1.1/search/tweets.json",
37   "GET",
38   $opt);
39
40 $result = json_decode($json);
41
42 $is_first = FALSE;
43 foreach($result->statuses as $status){
44   #text と created-at を入手して text に mecab を適用
45   $text = $status->text;
46   #$location = $status->user->location;
47   $create_at = $status->created_at;
48   $time = time() - (60*10);
49   $strtotime = strtotime($create_at);
50   if($strtotime < $time){return;}
51   //それ以外の場合（月日時が一致かつ差分が 10 分以内）のとき、以下を実行
52   for($node = $mecab->parseToNode($text); $node; $node = $node->getNext()){
53     if($node->getStat() != 2 && $node->getStat() != 3){
54       $feature = explode(",", $node->getFeature());
55       if(isset($feature[9])){
56         if(trim($feature[9]) === "1"){
57           $line[] = $feature[6];
58         }else if(trim($feature[9]) === "2"){
59           $station[] = $feature[6];
60         }
61       }
62     }
63   }
64 }
65 $line = array_merge(array_unique($line));
66 $station = array_merge(array_unique($station));
67 #DB と連携して路線名から駅名を引いてくる
68 try{
69   $dbh = new PDO($dsn,$usr,$pass);
70
71 #路線名から駅名検索して station にぶち込む
72 #路線名から路線 ID 入手$id[]へ

```



```

73 $sql = "select id from Line_memory where name= ?;";
74 $stmt = $dbh->prepare($sql);
75 foreach($line as $line_name){
76     $stmt->execute(array($line_name));
77     while($result = $stmt->fetch(PDO::FETCH_ASSOC)){
78         $id[]=$result['id'];
79     }
80 }
81 #id から駅名を引っ張ってきて station へ入れる
82 $sql = "select name from Station_memory where line_id= ?;";
83 $stmt = $dbh->prepare($sql);
84 foreach($id as $line_id){
85     $stmt->execute(array($line_id));
86     while($result = $stmt->fetch(PDO::FETCH_ASSOC)){
87         $station[] = $result['name'];
88     }
89 }
90
91 #station から WebSiteRanking, UserSearch を検索
92 $sql_wsr = "select id, dst from WebSiteRanking_memory where src = ?;";
93 $sql_us = "select id, dst from UserSearch_memory where src = ?;";
94 $wsr_id = array();
95 $us_id = array();
96 $stmt = $dbh->prepare($sql_wsr);
97 foreach($station as $from){
98     $stmt->execute(array($from));
99     while($result = $stmt->fetch(PDO::FETCH_ASSOC)){
100         $src[] = $from;
101         $dsts[] = $result['dst'];
102         $wsr_id[] = $result['id'];
103     }
104 }
105 $stmt = $dbh->prepare($sql_us);
106 foreach($station as $from){
107     $stmt->execute(array($from));
108     while($result = $stmt->fetch(PDO::FETCH_ASSOC)){
109         $src[] = $from;
110         $dsts[] = $result['dst'];
111         $us_id[] = $result['id'];
112     }
113 }
114
115 #更新処理
116 $update_wsr = "update WebSiteRanking set last_searched = now() where id = ?;";
117 $update_us = "update UserSearch set times = times + 1, last_searched = now where
id = ?;";
118
119 $stmt = $dbh->prepare($update_wsr);
120 foreach($wsr_id as $elm){

```

```

121     $stmt->execute(array($elm));
122 }
123 $stmt = $dbh->prepare($update_us);
124 foreach($sus_id as $elm){
125     $stmt->execute(array($elm));
126 }
127
128 }catch(PDOException $e){
129     print("database access failure: " . $e->getMessage());
130 }
131
132 #サーバ問い合わせ用クエリ生成して proxy 経由で問い合わせ
133
134 #問い合わせ用時刻設定（現在時刻を使用）
135 $Ym = date('Ym'); #yahoo jorudan year-month
136 $Yms = urlencode(date('Y/m')); #navitime year-month
137 $d = date('d'); #yahoo day
138 $j = date('j'); #jorudan navitime day
139 $H = date('H'); #yahoo hour
140 $G = date('G'); #jorudan navitime hour
141 $i = date('i');
142 $m10 = substr($i,0,1); #yahoo jorudan minute 10
143 $m1 = substr($i,1,1); #yahoo jorudan minute 1
144 if($i === "00"){
145     $mn = "0";
146 }else if(substr($i,0,1) === "0"){
147     $mn = substr($i,1,1);
148 }else{
149     $mn = $i;
150 }
151
152 #ループ回数設定
153 $length = count($src);
154 $QUERY = array();
155 for($i = 0; $i < 10; $i++){ //暫定 10 回
156     $urlenc_from = urlencode($src[$i]);
157     $urlenc_to = urlencode($dsts[$i]);
158
159     $y = "http://transit.loco.yahoo.co.jp/search/result?flatlon=&from=" .
    $urlenc_from . "&tlatlon=&to=" . $urlenc_to . "&via=&via=&via=&yym=" . $Ym . "&d=" .
    $d . "&hh=" . $H . "&m2=" . $m1 . "&m1=" . $m10 .
    "&type=1&ticket=ic&al=1&shin=1&ex=1&hb=1&lb=1&sr=1&s=0&expkind=1&ws=2&kw=" .
    $urlenc_to;
160
161     $jo = "http://www.jorudan.co.jp/norikae/cgi/nori.cgi?eki1=" . $urlenc_from .
    "&eki2=" . $urlenc_to . "&eki3=&via_on=1&Dym=" . $Ym . "&Ddd=" . $j . "&Dhh=" . $G .
    "&Dmn1=" . $m10 . "&Dmn2=" . $m1 .
    "&Cway=0&Cfp=1&C7=1&C2=0&C3=0&C1=0&C4=0&C6=2&S.x=10&S.y=15&S=%E6%A4%9C%E7%B4%A2
    &Cmap1=&rfr=nr&pg=0&eok1=&eok2=&eok3=&Csg=1";
162

```

```

163     $n     =     "http://www.navitime.co.jp/transfer/search?orvStationName=" .
$urlenc_from. "&dnvStationName=" . $urlenc_to . "&month=" . $Yms . "&day=" . $j .
"&hour=" . $G . "&minute=" . $mn .
"&basis=1&sort=0&wspeed=100&airplane=1&sprexprs=1&utrexprs=1&ferry=1&mtrplbus=1
&intercitybus=1&othexprs=1&ctl=020010&init=0&through=0";
164
165     array_push($QUERY,$y,$jo,$n);
166 }
167
168 //foreach($QUERY as $q){
169 //    print($q . "\n");
170 //}
171 //この処理別スレッドでやる必要があるかも
172 $hosts
array("transit.loco.yahoo.co.jp","www.jorudan.co.jp","www.navitime.co.jp"); =
173 $i = 0;
174 $threads = array();
175 foreach($QUERY as $getquery){
176     $threads[] = new GetData($getquery,$hosts[$i]);
177     $i++; if($i === 3){$i=0;}
178 }
179 $num = count($threads);
180 for($i=0; $i < $num; $i++){
181     $threads[$i]->start();
182 }
183
184 class GetData extends Thread{
185
186     private $host;
187     private $query;
188
189     function __construct($q, $h){
190         $this->query = $q;
191         $this->host = $h;
192     }
193
194     public function run(){
195
196         $header = array(
197             "Content-Type: application/x-www-form-urlencoded",
198             "Accept: /*/*",
199             "User-Agent: Mozilla/5.0",
200         );
201         $options = array(
202             "http" => array(
203                 "method" => "GET",
204                 "header" => implode("\r\n",$header),
205                 "proxy" => "tcp://localhost:8080",
206                 "request_fulluri" => true,
207             ),

```

```

208     );
209     file_get_contents($this->query,false,stream_context_create($options));
210 }
211 }
212 ?>

```

portal.php

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta http-equiv="Content-type" content="text/html; charset=utf-8">
5     <title>portal</title>
6     <script language="javascript">
7       /*global variables*/
8       /*for yahoo*/
9       var yfrom,yto,yvia1,yvia2,yvia3; //,yym,yd,yh,ym1,ym2; //出発到着経由、年月
        日時分
10      //var                                     ytype,
        yticket,yairplane,yshin,yexp,yhbus,ylbus,yferry,ysort,ykind,yspeed;
11      var                                     yselect_id =                                     new
        Array("yym","yd","yhh","ym1","ym2","ysort","yexpkind","yws");
12      var yselect_values = {};
13      var yradio_names = new Array("type","ticket");
14      var yradio_values = {};
15      var ycheck_id = new Array("yair","ysexp","yexp","yhbus","ybus","yfer");
16      var ycheck_values = {};
17      /*for jorudan*/
18      var jfrom,jto,jvia; //,jym,jd,jh,jm1,jm2; //出発到着経由、年月日時分
19      //var jtype, jticket,jft,jairplane,jhbus,jexp,jkind,jtime;
20      var                                     jselect_id =                                     new
        Array("jDym","jDdd","jDhh","jDmn1","jDmn2","jC7","jC2","jC3","jC1","jC4","jC6"
        );
21      var jselect_values = {};
22      var jradio_names = new Array("Cway","Cfp");
23      var jradio_values = {};
24      /*for navitime*/
25      var nfrom,nto,nvia1,nvia2,nvia3; //,nym,nd,nh,nm; //出発到着経由、年月日時
        分
26      //var ntype,nsort,nspeed,nairplane,nshin,nexp,nlbus,nothrexp,nferry;
27      var nselect_id = new Array("nmonth","nday","nhour","nminute");
28      var nselect_values = {};
29      var nradio_names = new Array("basis","sort","wspeed");
30      var nradio_values = {};

```

```

31     var                ncheck_id                =                new
Array("nairplane","nsprexprs","nutrexprs","nmtrplbus","nothexprs","nintercityb
us","nferry");
32     var ncheck_values = {};
33
34     window.onload = function(){
35     /*グローバル変数に値を入れる*/
36     var select;
37     var sopt;
38     var radio;
39     console.log(yselect_id[0]);
40     /*yahoo time*/
41     for(var i = 0; i < yselect_id.length; i++){
42         select = document.getElementById(yselect_id[i]); console.log(select);
43
44         sopt = document.getElementById(yselect_id[i]).options;
45         yselect_values[yselect_id[i]] = sopt.item(select.selectedIndex).value;
46
47         console.log(yselect_values[yselect_id[i]]);
48     }
49
50     for(var i = 0; i < yradio_names.length; i++){
51         radio = document.getElementsByName(yradio_names[i]);
52         for(var j = 0; j < radio.length; j++){
53             if(radio[j].checked){
54                 yradio_values[yradio_names[i]] = radio[j].value; break;
55             }
56         }
57     }
58
59     for(var i = 0; i < ycheck_id.length; i++){
60         if(document.getElementById(ycheck_id[i]).checked){
61             ycheck_values[ycheck_id[i]] =
62             document.getElementById(ycheck_id[i]).value;
63         }else{
64             ycheck_values[ycheck_id[i]] = "";
65         }
66     }
67
68     /*jorudan*/
69     for(var i = 0; i < jselect_id.length; i++){
70         select = document.getElementById(jselect_id[i]);
71         sopt = document.getElementById(jselect_id[i]).options;
72         jselect_values[jselect_id[i]] = sopt.item(select.selectedIndex).value;
73
74     }
75
76     for(var i = 0; i < jradio_names.length; i++){
77         radio = document.getElementsByName(jradio_names[i]);

```

```

73     for(var j = 0; j < radio.length; j++){
74         if(radio[j].checked){
75             jradio_values[jradio_names[i]] = radio[j].value; break;
76         }
77     }
78 }
79
80 /*navitime*/
81 for(var i = 0; i < nselect_id.length; i++){
82     select = document.getElementById(nselect_id[i]);
83     sopt = document.getElementById(nselect_id[i]).options;
84     nselect_values[nselect_id[i]] = sopt.item(select.selectedIndex).value;
85 }
86
87 for(var i = 0; i < nradio_names.length; i++){
88     radio = document.getElementsByName(nradio_names[i]);
89     for(var j = 0; j < radio.length; j++){
90         if(radio[j].checked){
91             nradio_values[nradio_names[i]] = radio[j].value; break;
92         }
93     }
94 }
95
96 for(var i = 0; i < ncheck_id.length; i++){
97     if(document.getElementById(ncheck_id[i]).checked){
98         ncheck_values[ncheck_id[i]] =
document.getElementById(ncheck_id[i]).value;
99     }else{
100         ncheck_values[ncheck_id[i]] = "";
101     }
102 }
103
104 }
105 /*引数 1:yahoo 2:jorudan 3:navitime*/
106 function PosttoProxy(host){
107     var param="";
108     var params = new Array();
109     if(host == 1){
110         yfrom = document.getElementById("yfrom").value;
111         yto = document.getElementById("yto").value;
112         yvia1 = document.getElementById("yvia1").value;
113         yvia2 = document.getElementById("yvia2").value;
114         yvia3 = document.getElementById("yvia3").value;
115         /*post用データ作成*/
116     }else if(host == 2){
117         jfrom = document.getElementById("jeki1").value;
118         jto = document.getElementById("jeki2").value;
119         jvia = document.getElementById("jeki3").value;

```

```

120     }else if(host == 3){
121         nfrom = document.getElementById("nfrom").value;
122         nto = document.getElementById("nto").value;
123         //nvia1 = document.getElementById("nvia1").value;
124         //nvia2 = document.getElementById("nvia2").value;
125         //nvia3 = document.getElementById("nvia3").value;
126     }
127     /*post 処理*/
128     var str = "";
129     if(host == 1){
130         str = "host=1&from=" + yfrom + "&to=" + yto + "&via1=" + yvia1 + "&via2="
+ yvia2 + "&via3=" + yvia3 + "&ym=" + yselect_values["ym"] + "&d=" +
yselect_values["yd"] + "&hh=" + yselect_values["yhh"] + "&m1=" +
yselect_values["ym1"] + "&m2=" + yselect_values["ym2"] + "&type=" +
yradio_values["type"] + "&ticket=" + yradio_values["ticket"] + "&al=" +
ycheck_values["yair"] + "&shin=" + ycheck_values["yexp"] + "&ex=" +
ycheck_values["yexp"] + "&hb=" + ycheck_values["yhbus"] + "&lb=" +
ycheck_values["ybus"] + "&sr=" + ycheck_values["yfer"] + "&s=" +
yselect_values["ysort"] + "&expkind=" + yselect_values["yexpkind"] + "&ws=" +
yselect_values["yws"] + "&kw=" + yfrom;
131     }else if(host == 2){
132         str = "host=2&from=" + jfrom + "&to=" + jto + "&via=" + jvia + "&Dym=" +
jselect_values["jDym"] + "&Ddd=" + jselect_values["jDdd"] + "&Dhh=" +
jselect_values["jDhh"] + "&Dmn1=" + jselect_values["jDmn1"] + "&Dmn2=" +
jselect_values["jDmn2"] + "&Cway=" + jradio_values["Cway"] + "&Cfp=" +
jradio_values["Cfp"] + "&C7=" + jselect_values["jC7"] + "&C2=" +
jselect_values["jC2"] + "&C3=" + jselect_values["jC3"] + "&C1=" +
jselect_values["jC1"] + "&C4=" + jselect_values["jC4"] + "&C6=" +
jselect_values["jC6"];
133     }else if(host == 3){
134         str = "host=3&from=" + nfrom + "&to=" + nto + "&via1=" + nvia1 + "&via2="
+ nvia2 + "&via3=" + nvia3 + "&month=" + nselect_values["nmonth"] + "&day=" +
nselect_values["nday"] + "&hour=" + nselect_values["nhour"] + "&minute=" +
nselect_values["nminute"] + "&basis=" + nradio_values["basis"] + "&sort=" +
nradio_values["sort"] + "&wspeed=" + nradio_values["wspeed"] + "&airplane=" +
ncheck_values["nairplane"] + "&sprexprs=" + ncheck_values["nsprexprs"] +
"&utrexprs=" + ncheck_values["nutrexprs"] + "&mtrplbus=" +
ncheck_values["nmtrplbus"] + "&othexprs=" + ncheck_values["nothexprs"] +
"&intercitybus=" + ncheck_values["nintercitybus"] + "&ferry=" +
ncheck_values["nferry"];
135     }
136     //document.getElementById("rewrite").innerHTML = str;
137     var xmlhttpRequest = new XMLHttpRequest();
138     xmlhttpRequest.onreadystatechange = function(){
139         var READYSTATE_COMPLETED = 4;
140         var HTTP_STATUS_OK = 200;
141
142         if( this.readyState == READYSTATE_COMPLETED && this.status ==
HTTP_STATUS_OK ){
143             // なにもしない
144             console.log( this.responseText );
145         }
146     }

```

```

147 xmlHttpRequest.open("POST", "http://192.168.225.190/pre_search_engine.php");
148     xmlHttpRequest.setRequestHeader('Content-Type',
'application/x-www-form-urlencoded' );
149     var encstr = encodeURI(str);
150     encstr.replace(/%20/g, '+');
151
152     xmlHttpRequest.send(encstr);
153     console.log("send" + encstr);
154 }
155 /*host 1:yahoo 2:jorudan 3:navitime*/
156 /*type 0:radio 1:select 2:checkbox*/
157 /*id:id*/
158 function ChangeData(host, type, id){
159     console.log("host: " + host + ",type: " + type + ",id: " + id);
160     if(host == 1){
161         if(type == 0){
162             var radio = document.getElementsByName(id);
163             for(var i = 0; i < radio.length; i++){
164                 if(radio[i].checked){
165                     yradio_values[id] = radio[i].value; break;
166                 }
167             }
168         }else if(type == 1){
169             var select = document.getElementById(id);
170             var sopt = document.getElementById(id).options;
171             yselect_values[id] = sopt.item(select.selectedIndex).value;
172         }else if(type == 2){
173             if(document.getElementById(id).checked){
174                 ycheck_values[id] = document.getElementById(id).value;
175             }else{
176                 ycheck_values[id] = "";
177             }
178         }
179     }else if(host == 2){
180         if(type == 0){
181             var radio = document.getElementsByName(id);
182             for(var i = 0; i < radio.length; i++){
183                 if(radio[i].checked){
184                     jradio_values[id] = radio[i].value; break;
185                 }
186             }
187         }else if(type == 1){
188             var select = document.getElementById(id);
189             var sopt = document.getElementById(id).options;
190             jselect_values[id] = sopt.item(select.selectedIndex).value;
191         }
192     }else if(host == 3){
193         if(type == 0){

```



```

194     var radio = document.getElementsByName(id);
195     for(var i = 0; i < radio.length; i++){
196         if(radio[i].checked){
197             nradio_values[id] = radio[i].value; break;
198         }
199     }
200     }else if(type == 1){
201         var select = document.getElementById(id);
202         var sopt = document.getElementById(id).options;
203         nselect_values[id] = sopt.item(select.selectedIndex).value;
204     }else if(type == 2){
205         if(document.getElementById(id).checked){
206             ncheck_values[id] = document.getElementById(id).value;
207         }else{
208             ncheck_values[id] = "";
209         }
210     }
211 }
212 }
213
214     /*onload の時の動作*/
215     </script>
216 </head>
217 <body>
218     <h1>Yahoo 乗り換え検索用検索窓</h1>
219     <form action="http://192.168.225.190/redirecttest.php" method="get"
name="search">
220         <input type="hidden" name="host" value="1" />
221         <div> 出発 :<input name="from" id="yfrom" type="text" value=""
onBlur="PosttoProxy(1)" /></div>
222         <div> 到着 :<input name="to" id="yto" type="text" value=""
onBlur="PosttoProxy(1)" /></div>
223         <div> 経由 1:<input name="via" id="yvia1" type="text" value=""
onBlur="PosttoProxy(1)" /></div>
224         <div> 経由 2:<input name="via" id="yvia2" type="text" value=""
onBlur="PosttoProxy(1)" /></div>
225         <div> 経由 3:<input name="via" id="yvia3" type="text" value=""
onBlur="PosttoProxy(1)" /></div>
226         <div>日時:
227         <select id="yym" name="ym" onchange="ChangeData(1,1,'yym')"
onBlur="PosttoProxy(1)">
228         <?php
229             $selected = "selected";
230             $year = date("Y");
231             $month = date("n");
232             $Y = $year;
233             for($j = 0; $j < 8; $j++){
234                 $M = $month + $j;

```

```

235         if($j == (12-$month)){ $Y = $Y + 1;}
236         if($M > 12){ $M = $M - 12;}
237         if($M < 10){ $Mm = "0" . $M;}else{$Mm = $M;}
238         echo '<option value="' . $Y . $Mm . "' ' . $selected . '>' . $Y . "
年" . $Mm . "月</option>\n";
239         //print("YM: " . $Y . $Mm . "\n");
240         $selected = "";
241     }
242     ?>
243 </select>
244 <select id="yd" name="d" onchange="ChangeData(1,1,'yd')"
onBlur="PosttoProxy(1)">
245     <?php
246         $j = date("j");
247         for($i = 1; $i <=31; $i++){
248             if($i < 10){$day = "0" . $i;}else{$day = $i;}
249             if($i == $j){
250                 echo '<option value="' . $day . '" selected>' . $i . "日</option>\n";
251             }else{
252                 echo '<option value="' . $day . "'>' . $i . "日</option>\n";
253             }
254         }
255     ?>
256 </select>
257 <select name="hh" id="yhh" onchange="ChangeData(1,1,'yhh')"
onBlur="PosttoProxy(1)">
258     <?php
259         $h = date("G");
260         for($i = 0; $i <= 23; $i++){
261             if($i < 10){ $hour = "0" . $i;}else{$hour = $i;}
262             if($i == $h){
263                 echo '<option value="' . $hour . '" selected>' . $i . "時
</option>\n";
264             }else{
265                 echo '<option value="' . $hour . "' >' . $i . "時</option>\n";
266             }
267         }
268     ?>
269 </select>
270 <select id="ym1" name="m1" onchange="ChangeData(1,1,'ym1')"
onBlur="PosttoProxy(1)">
271     <?php
272         $ii = date("i");
273         $m10 = substr($ii,0,1);
274         $m1 = substr($ii,1,1);
275         for($i =0; $i < 6; $i++){

```

```

276         if($i == $m10){
277             echo '<option value="" . $i . '" selected>' . $i . "</option>\n";
278         }else{
279             echo '<option value="" . $i . '" >' . $i . "</option>\n";
280         }
281     }
282
283     echo          "</select>\n<select          id='ym2'          name='m2'
onchange=\"ChangeData(1,1,'ym2')\" onBlur=\"PosttoProxy(1)\">\n";
284
285     for($i = 0; $i < 10; $i++){
286         if($i == $m1){
287             echo '<option value="" . $i . '" selected>' . $i . "</option>\n";
288
289             }else{
290                 echo '<option value="" . $i . '" >' . $i . "</option>\n";
291             }
292         }>
293     </select>分
294 </div>
295 <div>
296     <input type="radio" name="type" id="ytsDep" value="1" tabindex="700"
checked onchange="ChangeData(1,0,'type')" onBlur="PosttoProxy(1)" /><label
for="tsDep" >出発</label>
297     <input type="radio" name="type" id="ytsArr" value="4" tabindex="700"
onchange="ChangeData(1,0,'type')" onBlur="PosttoProxy(1)" /><label for="tsArr">
到着</label>
298     <input type="radio" name="type" id="ytsFir" value="3" tabindex="700"
onchange="ChangeData(1,0,'type')" onBlur="PosttoProxy(1)" /><label for="tsFir">
始発</label>
299     <input type="radio" name="type" id="ytsLas" value="2" tabindex="700"
onchange="ChangeData(1,0,'type')" onBlur="PosttoProxy(1)" /><label for="tsLas">
終電</label>
300     <input type="radio" name="type" id="ytsAvr" value="5" tabindex="700"
onchange="ChangeData(1,0,'type')" onBlur="PosttoProxy(1)" /><label for="tsAvr">
指定なし</label>
301 </div>
302 <div>運賃種別:
303     <label for="tic"><input type="radio" name="ticket" id="ytic" value="ic"
checked onchange="ChangeData(1,0,'ticket')" onBlur="PosttoProxy(1)" />IC カード
運賃優先</label>
304     <label for="tnormal"><input type="radio" name="ticket" id="ytnormal"
value="normal" onchange="ChangeData(1,0,'ticket')" onBlur="PosttoProxy(1)" />
現金運賃（きっぷ運賃）優先</label>
305 </div>
306 <div>交通手段
307     <input type="checkbox" name="al" id="yair" value="1" tabindex="630"
checked onchange="ChangeData(1,2,'yair')" onBlur="PosttoProxy(1)" />

```

```

308     <label for="air">空路</label>
309     <input type="checkbox" name="shin" id="ysexp" value="1" tabindex="630"
checked onchange="ChangeData(1,2,'ysexp')" onBlur="PosttoProxy(1)" />
310     <label for="sexp">新幹線</label>
311     <input type="checkbox" name="ex" id="yexp" value="1" tabindex="630"
checked onchange="ChangeData(1,2,'yexp')" onBlur="PosttoProxy(1)" />
312     <label for="exp">有料特急</label>
313     <input type="checkbox" name="hb" id="yhbus" value="1" tabindex="630"
checked onchange="ChangeData(1,2,'yhbus')" onBlur="PosttoProxy(1)" />
314     <label for="hbus">高速バス</label>
315     <input type="checkbox" name="lb" id="ybus" value="1" tabindex="630"
checked onchange="ChangeData(1,2,'ybus')" onBlur="PosttoProxy(1)" />
316     <label for="bus">路線バス</label>
317     <input type="checkbox" name="sr" id="yfer" value="1" tabindex="630"
checked onchange="ChangeData(1,2,'yfer')" onBlur="PosttoProxy(1)" />

318     <label for="fer">フェリー</label>
319 </div>
320 <div>表示順序:
321     <select name="s" id="ysort" onchange="ChangeData(1,1,'ysort')"
onBlur="PosttoProxy(1)">
322         <option value="0" selected>到着が早い順</option>
323         <option value="2" >乗り換え回数順</option>
324         <option value="1" >料金が安い順</option>
325     </select>
326     席指定:
327     <select name="expkind" id="yexpkind"
onchange="ChangeData(1,1,'yexpkind')" onBlur="PosttoProxy(1)">
328         <option value="1" selected>自由席優先</option>
329         <option value="2" >指定席優先</option>
330         <option value="3" >グリーン車優先</option>
331     </select>
332     歩く速度:
333     <select name="ws" id="yws" onchange="ChangeData(1,1,'yws')"
onBlur="PosttoProxy(1)">
334         <option value="1" >急いで</option>
335         <option value="2" selected>標準</option>
336         <option value="3" >少しゆっくり</option>
337         <option value="4" >ゆっくり</option>
338     </select>
339 </div>
340 <div> <input type="submit" value="検索" /></div>
341 </form>
342
343 <h1>ジョルダン乗り換え検索用窓</h1>
344 <form action="http://192.168.225.190/redirecttest.php" method="get"
name="fm_nori">

```

```

345     <input type="hidden" name="host" value="2" />
346     <div> 出発地 :<input type="text" name="eki1" id="jeki1" value=""
onBlur="PosttoProxy(2)" /></div>
347     <div> 到着地 :<input type="text" name="eki2" id="jeki2" value=""
onBlur="PosttoProxy(2)" /></div>
348     <div> 経由地 :<input type="text" name="eki3" id="jeki3" value=""
onBlur="PosttoProxy(2)" /></div>
349     <input type="hidden" name="via_on" value="1" />
350     <div>日時:
351     <select name="Dym" id="jDym" onchange="ChangeData(2,1,'jDym')"
onBlur="PosttoProxy(2)">
352         <?php
353             $selected = "selected";
354             $year = date("Y");
355             $month = date("n");
356             $Y = $year;
357             for($j = 0; $j < 8; $j++){
358                 $M = $month + $j;
359                 if($j == (12-$month)){ $Y = $Y + 1;}
360                 if($M > 12){ $M = $M - 12;}
361                 if($M < 10){ $Mm = "0" . $M;}else{$Mm = $M;}
362                 echo '<option value="' . $Y . $Mm . "' ' . $selected . '>' . $Y . "
年" . $Mm . "月</option>\n";
363                 //print("YM: " . $Y . $Mm . "\n");
364                 $selected = "";
365             }
366         ?>
367     </select>
368     <select id="jDdd" name="Ddd" size="1" tabindex="2"
onchange="ChangeData(2,1,'jDdd')" onBlur="PosttoProxy(2)">
369     <?php
370         $j = date("j");
371         for($i = 1; $i <=31; $i++){
372             if($i == $j){
373                 echo '<option value="' . $i . "' selected>' . $i . "日</option>\n";
374             }else{
375                 echo '<option value="' . $i . "'>' . $i . "日</option>\n";
376             }
377         }
378     ?>
379     </select>
380     <select id="jDhh" name="Dhh" size="1" tabindex="2"
onchange="ChangeData(2,1,'jDhh')" onBlur="PosttoProxy(2)">
381     <?php
382         $h = date("G");
383         for($i = 0; $i <= 23; $i++){
384             if($i == $h){

```

```

385         echo '<option value="" . $i . ' selected>' . $i . "時</option>";
386     }else{
387         echo '<option value="" . $i . ' >' . $i . "時</option>\n";
388     }
389 }
390 ?>
391 </select>時
392 <select id="jDmn1" name="Dmn1" size="1" tabindex="2"
onchange="ChangeData(2,1,'jDmn1')" onBlur="PosttoProxy(2)">
393 <?php
394     $ii = date("i");
395     $m10 = substr($ii,0,1);
396     $m1 = substr($ii,1,1);
397     for($i =0; $i < 6; $i++){
398         if($i == $m10){
399             echo '<option value="" . $i . ' selected>' . $i . "</option>";
400
401         }else{
402             echo '<option value="" . $i . ' >' . $i . "</option>";
403         }
404     }
405     echo "</select>\n<select id='jDmn2' name='Dmn2' size='1' tabindex='2'
onchange=\"ChangeData(2,1,'jDmn2')\" onBlur=\"PosttoProxy(2)\">\n";
406
407     for($i = 0; $i < 10; $i++){
408         if($i == $m1){
409             echo '<option value="" . $i . ' selected>' . $i . "</option>\n";
410         }else{
411             echo '<option value="" . $i . ' >' . $i . "</option>\n";
412         }
413     }
414
415 ?>
416 </select>
417 分
418 </div>
419 <div>
420     <input type="radio" id="jCway0" name="Cway" value="0" tabindex="3"
class="rd" checked="checked" onchange="ChangeData(2,0,'Cway')"
onBlur="PosttoProxy(2)" /><label for="Cway0">出発</label>
421     <input type="radio" id="jCway1" name="Cway" value="1" tabindex="3"
class="rd" onchange="ChangeData(2,0,'Cway')" onBlur="PosttoProxy(2)" /><label
for="Cway1">到着</label>
422     <input type="radio" id="jCway2" name="Cway" value="2" tabindex="3"
class="rd" onchange="ChangeData(2,0,'Cway')" onBlur="PosttoProxy(2)" /><label
for="Cway2">始発</label>

```

```

423     <input type="radio" id="jCway3" name="Cway" value="3" tabindex="3"
class="rd" onchange="ChangeData(2,0,'Cway')" onBlur="PosttoProxy(2)"/><label
for="Cway3">終電</label>
424     </div>
425     <div>運賃:
426     <input type="radio" id="jCfp1" name="Cfp" value="1" tabindex="3"
class="rd" checked="checked" onchange="ChangeData(2,0,'Cfp')"
onBlur="PosttoProxy(2)"/><label for="Cfp1">IC カード利用</label>
427     <input type="radio" id="jCfp2" name="Cfp" value="2" tabindex="3"
class="rd" onchange="ChangeData(2,0,'Cfp')" onBlur="PosttoProxy(2)"/><label
for="Cfp2">切符利用</label>
428     </div>
429     <div>条件</div>
430     <div>定期の種類:
431     <select id="jC7" name="C7" size="1" tabindex="4"
onchange="ChangeData(2,1,'jC7')" onBlur="PosttoProxy(2)">
432         <option value="1" selected="selected">通勤</option>
433         <option value="2" >通学 (大学生) </option>
434         <option value="3" >通学 (高校生) </option>
435         <option value="4" >通学 (中学生) </option>
436     </select>
437     </div>
438     <div>飛行機:
439     <select id="jC2" name="C2" size="1" tabindex="4"
onchange="ChangeData(2,1,'jC2')" onBlur="PosttoProxy(2)">
440         <option value="0" selected="selected" >おまかせ</option>
441         <option value="1" >使わない</option>
442     </select>
443     </div>
444     <div>高速バス:
445     <select id="jC3" name="C3" size="1" tabindex="4"
onchange="ChangeData(2,1,'jC3')" onBlur="PosttoProxy(2)">
446         <option value="0" selected="selected" >おまかせ</option>
447         <option value="1" >使わない</option>
448     </select>
449     </div>
450     <div>有料特急:
451     <select id="jC1" name="C1" size="1" tabindex="4"
onchange="ChangeData(2,1,'jC1')" onBlur="PosttoProxy(2)">
452         <option value="0" selected="selected" >おまかせ</option>
453         <option value="1" >使わない</option>
454         <option value="2" >100km 以上なら使う</option>
455     </select>
456     </div>
457     <div>優先座席:
458     <select id="jC4" name="C4" size="1" tabindex="4"
onchange="ChangeData(2,1,'jC4')" onBlur="PosttoProxy(2)">
459         <option value="0" selected="selected" >指定席</option>
460         <option value="1" >自由席</option>

```

```

461         <option value="2" >グリーン席</option>
462     </select>
463 </div>
464 <div>乗換時間:
465     <select      id="jC6"      name="C6"      size="1"      tabindex="4"
onchange="ChangeData(2,1,'jC6')" onBlur="PosttoProxy(2)">
466         <option value="1" >短め</option>
467         <option value="2" selected="selected" >標準</option>
468         <option value="3" >余裕を持つ</option>
469     </select>
470 </div>
471 <div> <input type="submit" value="検索" /></div>
472 <input type="hidden" name="S" value="検索" />
473 <input type="hidden" name="Cmap1" value="" />
474 <input type="hidden" name="rf" value="nr" />
475 <input type="hidden" name="pg" value="0" />
476 <input type="hidden" name="eok1" value="" />
477 <input type="hidden" name="eok2" value="" />
478 <input type="hidden" name="eok3" value="" />
479 <input type="hidden" name="Csg" value="1" />
480 </form>
481 <h1>NAVITIME 検索用窓</h1>
482 <form action="http://www.navitime.co.jp/transfer/search" method="GET">
483     <input type="hidden" name="host" value="3" />
484     <div> 乗車駅 :<input type="text" name="orvStationName" id="nfrom"
onBlur="PosttoProxy(3)" /></div>
485     <div> 降車駅 :<input type="text" name="dnvStationName" id="nto"
onBlur="PosttoProxy(3)" /></div>
486     <!--
487     <div> 経由駅 1:<input type="text" name="thrStationName1" id="nvia1"
onBlur="PosttoProxy(3)" /></div>
488     <div> 経由駅 2:<input type="text" name="thrStationName2" id="nvia2"
onBlur="PosttoProxy(3)" /></div>
489     <div> 経由駅 3:<input type="text" name="thrStationName3" id="nvia3"
onBlur="PosttoProxy(3)" /></div>
490     -->
491     <div>日時:
492     <select name="month" id="nmonth" onchange="ChangeData(3,1,'nmonth')"
onBlur="PosttoProxy(3)">
493     <?php
494         $selected = "selected";
495         $year = date("Y");
496         $month = date("n");
497         $Y = $year;
498         for($j = 0; $j < 8; $j++){
499             $M = $month + $j;
500             if($j == (12-$month)){ $Y = $Y + 1;}
501             if($M > 12){ $M = $M - 12;}

```



```

502         if($M < 10){ $Mm = "0" . $M;}else{$Mm = $M;}
503         echo '<option value="' . $Y . '/' . $Mm . "' ' . $selected . '>' . $Y .
"年" . $Mm . "</option>\n";
504         //print("YM: " . $Y . $Mm . "\n");
505         $selected = "";
506     }
507     ?>
508     </select>月
509     <select name="day" id="nday"onchange="ChangeData(3,1,'nday')"
onBlur="PosttoProxy(3)">
510         <?php
511         $j = date("j");
512         for($i = 1; $i <=31; $i++){
513             if($i < 10){$day = "0" . $i;}else{$day = $i;}
514             if($i == $j){
515                 echo '<option value="' . $day . "' selected>' . $i . "</option>\n";
516             }else{
517                 echo '<option value="' . $day . "'>' . $i . "</option>\n";
518             }
519         }
520         ?>
521     </select>日
522     <select name="hour" id="nhour" onchange="ChangeData(3,1,'nhour')"
onBlur="PosttoProxy(3)">
523         <?php
524         $h = date("G");
525         for($i = 0; $i <= 23; $i++){
526             if($i == $h){
527                 echo '<option value="' . $i . "' selected>' . $i . "</option>\n";
528             }else{
529                 echo '<option value="' . $i . "' >' . $i . "</option>\n";
530             }
531         }
532         ?>
533     </select>時
534     <select name="minute" id="nminute" onchange="ChangeData(3,1,'nminute')"
onBlur="PosttoProxy(3)">
535         <?php
536         $m = date("i");
537         for($i = 0; $i < 60; $i++){
538             if($i == $m){
539                 echo '<option value="' . $i . "' selected>' . $i . "</option>\n";
540             }else{
541                 echo '<option value="' . $i . "' >' . $i . "</option>\n";
542             }
543         }

```

```

544     ?>
545     </select>分
546 </div>
547 <!--
548 <div>
549     <input type="radio" name="basis" value="1" checked="checked"
id="nbasis1" onchange="ChangeData(3,0,'basis')" onBlur="PosttoProxy(3)" />出発
550     <input type="radio" name="basis" value="0" id="nbasis2"
onchange="ChangeData(3,0,'basis')" onBlur="PosttoProxy(3)" />到着
551     <input type="radio" name="basis" value="4" id="nbasis3"
onchange="ChangeData(3,0,'basis')" onBlur="PosttoProxy(3)" />始発
552     <input type="radio" name="basis" value="3" id="nbasis4"
onchange="ChangeData(3,0,'basis')" onBlur="PosttoProxy(3)" />終電
553 </div>
554 -->
555 <div>表示順序
556     <input type="radio" value="0" name="sort" checked id="nsort1"
onchange="ChangeData(3,0,'sort')" onBlur="PosttoProxy(3)" />時間
557     <input type="radio" value="1" name="sort" id="nsort2"
onchange="ChangeData(3,0,'sort')" onBlur="PosttoProxy(3)" />運賃
558     <input type="radio" value="2" name="sort" id="nsort3"
onchange="ChangeData(3,0,'sort')" onBlur="PosttoProxy(3)" />乗換回数
559     <input type="radio" value="6" name="sort" id="nsort4"
onchange="ChangeData(3,0,'sort')" onBlur="PosttoProxy(3)" />CO2 排出量
560 </div>
561 <div>
562     <input type="radio" value="7" name="sort" id="nsort5"
onchange="ChangeData(3,0,'sort')" onBlur="PosttoProxy(3)" />エレベーター優先
563     <input type="radio" value="8" name="sort" id="nsort6"
onchange="ChangeData(3,0,'sort')" onBlur="PosttoProxy(3)" />エレベーター・エス
カレーター優先
564 </div>
565 <div>徒歩速度:
566     <input type="radio" value="100" name="wspeed" id="nwspeed1" checked
onchange="ChangeData(3,0,'wspeed')" onBlur="PosttoProxy(3)" />標準
567     <input type="radio" value="125" name="wspeed" id="nwspeed2"
onchange="ChangeData(3,0,'wspeed')" onBlur="PosttoProxy(3)" />ゆっくり
568     <input type="radio" value="83" name="wspeed" id="nwspeed3"
onchange="ChangeData(3,0,'wspeed')" onBlur="PosttoProxy(3)" />せかせか
569 </div>
570 <div>利用路線:
571     <input type="checkbox" name="airplane" value="1" checked id="nairplane"
onchange="ChangeData(3,2,'nairplane')" onBlur="PosttoProxy(3)" />飛行機
572     <input type="checkbox" name="sprexprs" value="1" checked id="nsprexprs"
onchange="ChangeData(3,2,'nsprexprs')" onBlur="PosttoProxy(3)" />新幹線
573     <input type="checkbox" name="utrexprs" value="1" checked id="nutrexprs"
onchange="ChangeData(3,2,'nutrexprs')" onBlur="PosttoProxy(3)" />特急線
574     <input type="checkbox" name="mtrplbus" value="1" checked id="nmtrplbus"
onchange="ChangeData(3,2,'nmtrplbus')" onBlur="PosttoProxy(3)" />路線バス
575 </div>
576 <div>

```

```

577     <input type="checkbox" name="othexprs" value="1" id="nothexprs" checked
onchange="ChangeData(3,2,'nothexprs')" onBlur="PosttoProxy(3)" />その他有料路線
578     <input type="checkbox" name="intercitybus" value="1" id="nintercitybus"
checked onchange="ChangeData(3,2,'nintercitybus')" onBlur="PosttoProxy(3)" />
高速バス
579     <input type="checkbox" name="ferry" value="1" id="nferry" checked
onchange="ChangeData(3,2,'nferry')" onBlur="PosttoProxy(3)" />フェリー
580     </div>
581     <input type="hidden" id="transfer_orv_code" name="orvStationCode" value=""
/>
582     <input type="hidden" id="transfer_dnv_code" name="dnvStationCode" value=""
/>
583
584     <div>
585         <input type="radio" name="basis" value="1" checked="checked"
id="nbasis1" onchange="ChangeData(3,0,'basis')" onBlur="PosttoProxy(3)" />出発
586         <input type="radio" name="basis" value="0" id="nbasis2"
onchange="ChangeData(3,0,'basis')" onBlur="PosttoProxy(3)" />到着
587         <input type="radio" name="basis" value="4" id="nbasis3"
onchange="ChangeData(3,0,'basis')" onBlur="PosttoProxy(3)" />始発
588         <input type="radio" name="basis" value="3" id="nbasis4"
onchange="ChangeData(3,0,'basis')" onBlur="PosttoProxy(3)" />終電
589     </div>
590     <input type="hidden" id="transfer_ctl" name="ctl" value="020010" />
591     <input type="hidden" value="2" name="atr" />
592     <input type="hidden" value="1" name="detailset" />
593     <input type="hidden" id="transfer_init" name="init" class="dynamic" />
594     <input type="submit" value="検索" />
595 </form>
596
597     <div id="rewrite"> かきかえるよー </div>
598 </body>
599 </html>

```

redirector.php

```

1  <?php
2  $host = $_GET["host"];
3  /**usersearch にカラム追加など処理***/
4  /*src dst 駅を入手*/
5  if($host == 1){
6  $src = $_GET["from"];
7  $dst = $_GET["to"];
8  }else if($host == 2){
9  $src = $_GET["eki1"];
10 $dst = $_GET["eki2"];
11 }else{
12 $src = $_GET["orvStationName"];
13 $dst = $_GET["dnvStationName"];
14 }

```

```

15 /*dbへ接続*/
16 $dsn = "mysql:host=localhost;dbname=transit;charset=utf8";
17 $usr = "yonemura";
18 $pass = "ekripssoft";
19 $dbh = new PDO($dsn,$usr,$pass);
20 /*先読み中の場合はここで先読みが終了するまで待つ?*/
21 /*location heaer を返す*/
22 $host = $_GET["host"];
23 if($host == 1){
24 $location = "http://transit.loco.yahoo.co.jp/search/result?from=" .
urlencode($_GET["from"]) . "&to=" . urlencode($_GET["to"]) . "&via=" .
urlencode($_GET["via1"]) . "&via=" . urlencode($_GET["via2"]) . "&via=" .
urlencode($_GET["via3"]) . "&ym=" . $_GET["ym"] . "&d=" . $_GET["d"] .
"&hh=" . $_GET["hh"] . "&m1=" . $_GET["m1"] . "&m2=" . $_GET["m2"] . "&type=" .
$_GET["type"] . "&ticket=" . $_GET["ticket"] . "&a1=" . $_GET["a1"] .
"&shin=" . $_GET["shin"] . "&ex=" . $_GET["ex"] . "&hb=" . $_GET["hb"] .
"&lb=" . $_GET["lb"] . "&sr=" . $_GET["sr"] . "&s=" . $_GET["s"] .
"&expkind=" . $_GET["expkind"] . "&ws=" . $_GET["ws"] . "&kw=" .
urlencode($_GET["to"]);
25 }else if($host == 2){
26 $location = "http://www.jorudan.co.jp/norikae/cgi/nori.cgi?eki1=" .
urlencode($_GET["eki1"]) . "&eki2=" . urlencode($_GET["eki2"]) . "&eki3=" .
urlencode($_GET["eki3"]) . "&via_on=" . $_GET["via_on"] . "&Dym=" .
$_GET["Dym"] . "&Ddd=" . $_GET["Ddd"] . "&Dhh=" . $_GET["Dhh"] . "&Dmn1=" .
$_GET["Dmn1"] . "&Dmn2=" . $_GET["Dmn2"] . "&Cway=" . $_GET["Cway"] .
"&Cfp=" . $_GET["Cfp"] . "&C7=" . $_GET["C7"] . "&C2=" . $_GET["C2"] . "&C3=" .
$_GET["C3"] . "&C1=" . $_GET["C1"] . "&C4=" . $_GET["C4"] . "&C6=" .
$_GET["C6"] .
"&S=%E6%A4%9C%E7%B4%A2&Cmap1=&rfr=nr&pg=0&eok1=&eok2=&eok3=&Csg=1";
27 }else if($host == 3){
28 $location = "http://www.navitime.co.jp/transfer/search?orvStationName=" .
urlencode($_GET["orvStationName"]) . "&dnvStationName=" .
urlencode($_GET["dnvStationName"]) . "&month=" .
urlencode($_GET["month"]) . "&day=" . $_GET["day"] . "&hour=" .
$_GET["hour"] . "&minute=" . $_GET["minute"] . "&sort=" . $_GET["sort"] .
"&wspeed=" . $_GET["wspeed"] . "&airplane=" . $_GET["airplane"] .
"&sprexprs=" . $_GET["sprexprs"] . "&utrexprs=" . $_GET["utrexprs"] .
"&mtrplbus=" . $_GET["mtrplbus"] . "&othexprs=" . $_GET["othexprs"] .
"&intercitybus=" . $_GET["intercitybus"] . "&ferry=" . $_GET["ferry"] .
"&orvStationCode=&dnvStationCode=&basis=" . $_GET["basis"] .
"&ctl=020010&atr=2&detailset=1&init=";
29 }
30 header("Location:" . $location);
31
32 try{
33 // $dbh = new PDO($dsn,$usr,$pass);
34 $query = "select count(*) from UserSearch_memory where src=\"\" . $src . "\"
and dst=\"\" . $dst . "\"";
35 $stmt = $dbh->query($query);
36 $us_rows = $stmt->fetchColumn();
37 $stmt = null;
38 $query = "select count(*) from WebSiteRanking_memory where src=\"\" . $src .
\"\" and dst=\"\" . $dst . "\"";
39 $stmt = $dbh->query($query);

```

```

40 $wsr_rows = $stmt->fetchColumn();
41 $stmt=null;
42 if(($us_rows + $wsr_rows) == 0){//UserSearch WebSiteRanking とともにないとき
43 $sql = "insert into UserSearch_memory (src,dst,times,last_searched,host)
values(?,?,1,now()," . $host . ")";
44 $stmt = $dbh->prepare($sql);
45 $flag = $stmt->execute(array($src,$dst));
46 }else{
47 $id = array();
48 if($us_rows > 0){
49 $query = "select id from UserSearch_memory where src=\"\" . $src . "\" and
dst=\"\" . $dst . "\"";
50 $stmt= $dbh->query($query);
51 while($row = $stmt->fetch(PDO::FETCH_ASSOC)){
52 $id[] = $row["id"];
53 }
54 $query = "update UserSearch_memory set last_searched=now(),times = times+1
where id=?";
55 $stmt = $dbh->prepare($query);
56 foreach($id as $elm){
57 $stmt->execute(array($elm));
58 }
59 }else if($wsr_rows > 0){
60 $query = "select id from WebSiteRanking_memory where src=\"\" . $src . "\" and
dst=\"\" . $dst . "\"";
61 $stmt=$dbh->query($query);
62 while($row = $stmt->fetch(PDO::FETCH_ASSOC)){
63 $id[] = $row["id"];
64 }
65 $query = "update WebSiteRanking_memory set last_searched=now(),times =
times+1 where id=?";
66 $stmt = $dbh->prepare($query);
67 foreach($id as $elm){
68 $stmt->execute(array($elm));
69 }
70 }
71 }
72 }catch(PDOException $e){
73 print("error:" . $e->getMessage());
74
75 }
76 ?>

```

付録 C 各種設定

先読みサーバに使用した apache の設定を付する。また、apache と php はソースコードからコンパイルしたものを利用した。コンパイルする際に設定した内容を付する。

apache の設定(httpd.conf)抜粋

...	
86	LoadModule cache_module modules/mod_cache.so
87	LoadModule cache_disk_module modules/mod_cache_disk.so
116	LoadModule proxy_module modules/mod_proxy.so
117	LoadModule proxy_connect_module modules/mod_proxy_connect.so
118	LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
119	LoadModule proxy_http_module modules/mod_proxy_http.so
120	LoadModule proxy_fcgi_module modules/mod_proxy_fcgi.so
121	LoadModule proxy_scgi_module modules/mod_proxy_scgi.so
122	LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
123	LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
124	LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
125	LoadModule proxy_express_module modules/mod_proxy_express.so
150	LoadModule php5_module modules/libphp5.so
151	LoadModule fcgid_module modules/mod_fcgid.so
...	
167	ExtendedStatus On
168	<Location /server-status>
169	SetHandler server-status
170	Order Deny,Allow
171	Deny from all
172	Allow from 127.0.0.1
173	Allow from localhost
174	Allow from 192.168.225
175	</Location>
176	<Location /server-info>
177	SetHandler server-info
178	Order Deny,Allow
179	Allow from 127.0.0.1
180	Allow from localhost
181	Allow from 192.168.225
182	</Location>
...	
311	<IfModule log_config_module>
312	#
313	# The following directives define some format nicknames for use with

```

314     # a CustomLog directive (see below).
315     #
316     LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
317     LogFormat "%h %l %u %t \"%r\" %>s %b" common
318     LogFormat "%h %{canonical}p %{local}p %{remote}p %l %u %t \"%r\" %>s %b
\"%{Referer}i\" \"%{User-Agent}i\"" proxylog
319     <IfModule logio_module>
320         # You need to enable mod_logio.c to use %I and %O
321         LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\"
\"%{User-Agent}i\" %I %O" combinedio
322     </IfModule>
...
547 ProxyRequests On
548 ProxyVia On
549 #ErrorLog logs/proxy_error_log combined
550 #CustomLog logs/proxy_log combined
551 <Proxy *:8080>
552     Order deny,allow
553     Deny from all
554     Allow from 192.168.225. localhost
555     #Header set Expires "Wed Thu, 01 Jan 2099 00:00:00 GMT"
556 </Proxy>
557
558 <VirtualHost *:8080>
559
560 ErrorLog logs/proxy_error_log
561 CustomLog logs/proxy_log proxylog
562 #<IfModule Proxy_Module>
563     #ProxyRequests On
564     #ProxyVia On
565     #CustomLog logs/proxy_log combined
566     #<Proxy *>
567     #     Order deny,allow
568     #     Deny from all
569     #     Allow from 192.168.225. localhost
570     #</Proxy>
571     #<IfModule mod_cache.c>
572         #<IfModule mod_disk_cache.c>
573             LogFormat "%t %h %u \"%r\" \"%{cache-status}e\"" cachelog
574             CustomLog logs/customcache.log cachelog
575             CustomLog logs/cached-requests.log combined env=cache-hit
576             CustomLog logs/uncached-requests.log combined env=cache-miss

```

```

577 CustomLog logs/revalidated-requests.log combined
env=cache-revalidate
578 CustomLog logs/invalidate-requests.log combined
env=cache-invalidate
579 CustomLog logs/yahootransit.log combined env=yahootr
580 CustomLog logs/jorudantransit.log combined env=jorudantr
581 CustomLog logs/navitimetransit.log combined env=navitimetr
582 CacheRoot /var/cache/httpd
583 #CacheSize 50000
584 CacheEnable disk http://transit.loco.yahoo.co.jp/
585 CacheEnable disk http://www.jorudan.co.jp/
586 CacheEnable disk http://www.navitime.co.jp/
587 CacheEnable disk http://192.168.225.191/
588 CacheDirLevels 1
589 CacheDirLength 2
590 CacheIgnoreCacheControl on
591 CacheIgnoreNoLastmod on
592 CacheStoreNoStore on
593 CacheStorePrivate on
594 #CacheDefaultExpire 3600
595 CacheIgnoreHeaders Set-Cookie User-Agent
596
597 Header set Expires "Wed, 01 Apr 2015 00:00:00 GMT"
598 Header unset Vary
599 #AllowEncodedSlashes on
600 #</IfModule>
601 #</IfModule>
602 </VirtualHost>
603 AllowEncodedSlashes on

```

apache コンパイル時設定

```

1 #! /bin/sh
2 #
3 # Created by configure
4
5 ./configure \
6 --with-apr=/usr/local/apr-httpd/ \
7 --with-apr-util=/usr/local/apr-util-httpd/ \
8 --enable-cache \
9 --enable-disk-cache \
10 --enable-headers \
11 --enable-mem-cache \

```



```
12 --enable-proxy \  
13 --enable-proxy-connect \  
14 --enable-proxy-http \  
15 --enable-rewrite \  
16 --enable-so \  
17 $@
```

php コンパイル時設定

```
1 #!/bin/sh  
2 #  
3 # Created by configure  
4  
5 './configure' \  
6 '--with-apxs2=/usr/local/apache2/bin/apxs' \  
7 '--with-mysql' \  
8 '--with-mysqli' \  
9 '--with-pdo-mysql' \  
10 '--with-pear' \  
11 '--with-curl' \  
12 '--enable-maintainer-zts' \  
13 '--with-zlib' \  
14 '--enable-mbstring' \  
15 '--with-bz2' \  
16 '--enable-zip' \  
17 '--enable-sockets' \  
18 '--enable-sysvsem' \  
19 '--enable-sysvshm' \  
20 '--with-mhash' \  
21 '--with-pcre-regex' \  
22 '--with-gettext' \  
23 '--enable-bcmath' \  
24 '--enable-libxml' \  
25 '--enable-json' \  
26 '--with-openssl' \  
27 '--enable-pcntl' \  
28 '--enable-fpm' \  
29 $@
```

付録 D 評価用シナリオ，スクリプト等

本節では，評価に利用したシナリオを作成するスクリプト，先読みサーバに問い合わせを行うスクリプトなどを記載する。

なお，シナリオを作成する際に利用した国土交通省の大都市交通センサスは，報告書資料のうち初乗り・最終降車駅間移動人員表，および初乗り最終降車駅間経路別人員表の二つの資料に記載されている駅を利用して，移動人員を算出した。

駅間移動人員と検索回数を保持するテキスト（抜粋）

```
東京,品川,381,22
東京,川崎,228,13
東京,横浜,372,21
東京,平塚,114,7
東京,桜木町,252,15
東京,新横浜,60,4
東京,品川シーサイド,234,14
新橋,品川,90,6
新橋,川崎,114,7
新橋,熊谷,63,4
品川,東京,1689,95
品川,新橋,711,40
品川,川崎,240,14
品川,横浜,351,20
品川,戸塚,225,13
品川,藤沢,36,3
品川,四ッ谷,498,28
品川,恵比寿,1029,58
品川,武蔵中原,45,3
品川,八丁堀,87,5
品川,赤坂見附,393,23
品川,溜池山王,54,4
品川,銀座,180,11
品川,日本橋,336,19
品川,三越前,138,8
品川,本郷三丁目,231,13
品川,九段下,114,7
品川,竹橋,126,8
品川,木場,591,34
品川,東陽町,96,6
品川,本厚木,63,4
品川,南林間,180,11
…以下 77592 組まで続く
```

*乗車駅,降車駅,移動人員,検索回数の順に記載

シナリオ作成用スクリプト（make_scenario.php）

```
1 <?php
2 error_reporting(E_ALL ^ E_NOTICE);
3 $fp = fopen("allstations_per","r");
4 $hour =
  array(30000,50000,62000,61000,62000,62000,62000,60000,58000,59000,62000,
```

```

5         62000,61000,60000,58000,60000,70000,79000,59000,30000);
6 $key_ary = array();
7 $station = array();
8
9 while(($elm =fgetcsv($fp,",")) !== FALSE){
10     $key = $elm[0] . "," . $elm[1];
11     $key_ary[] = $key;
12     $station[$key] = str_replace(array("\r\n","\r","\n"),",",$elm[3]);
13 }
14 /*start create scenario*/
15 for($i=0;$i<20;$i++){
16     print("_____\n");
17     for($j=0;$j<$hour[$i];$j++){
18         $count = count($key_ary);
19         $key_num = rand(0,$count);
20         $key = $key_ary[$key_num];
21         print($key . "\n");
22         $station[$key] -= 1;
23         if($station[$key] == 0){
24             unset($key_ary[$key_num]);
25             unset($station[$key]);
26             $key_ary = array_merge($key_ary);
27         }
28     }
29 }
30 }
31 ?>

```

ソーシャル情報を利用したシナリオ-クライアント側 (create_scenario_w_social_client.php)

取得したソーシャル情報は sep ディレクトリに, "時間 (分), 遅延路線" と書かれたファイルに存在している. また, 駅間移動人員と検索回数の情報はデータベースにも存在しているとする.

```

1 <?php
2 #argv[1]:遅延情報ファイル,argv[2]:書き込み先ファイル,$argv[3]:最大個
  数,argv[4]:乗車駅数制限
3 $file="./sep/".$argv[1];
4 $fp=fopen($file,"r");
5 $wfp = fopen($argv[2],"w");
6 $dsn = 'mysql:host=localhost;dbname=transit;charset=utf8';
7 $usr = 'yonemura';
8 $pass = 'ekripssoft';
9 $dbh = new PDO($dsn,$usr,$pass);
10
11 while(($data=fgetcsv($fp,",")) !== FALSE){

```

```

12     fwrite($wfp,"___".$data[0]."\n");
13     $line = $data[1];
14     $id = array();
15     $station = array();
16     $max = $argv[3];
17     try{
18         $sql = "select id from Line_memory where name= ?;";
19         $stmt = $dbh->prepare($sql);
20         $stmt->execute(array($line));
21         while($result = $stmt->fetch(PDO::FETCH_ASSOC)){
22             $id[]=$result['id'];
23         }
24
25         #idから駅名を引っ張ってきてstationへ入れる
26         #この数が爆発するので、制限する。
27         $sql = "select name from Station_memory where line_id= ? order by
src_popularity desc limit " . $argv[4] . " . ";";
28         $stmt = $dbh->prepare($sql);
29         foreach($id as $line_id){
30             $stmt->execute(array($line_id));
31
32             while($result = $stmt->fetch(PDO::FETCH_ASSOC)){
33                 $station[] = $result['name'];
34             }
35             $dsts = (int)$max/$argv[4];
36             /*Station名からCensus_memoryを引いて駅を選ぶ。各駅につき上位をラ
ンダム個。回数も上位から*/
37             $sql = "select dst from Census_memory where src=? order by num desc
limit ".$dsts." . ";";
38             $stmt = $dbh->prepare($sql);
39             foreach($station as $src){
40                 $stmt->execute(array($src));
41
42                 while($result = $stmt->fetch(PDO::FETCH_ASSOC)){
43                     $rand=mt_rand(1,10);
44                     for($i=0;$i<$rand;$i++){
45                         fwrite($wfp,$src.", ".$result['dst']."\n");
46                     }
47                 }
48             }catch(PDOException $e){
49                 print($e->getMessage());
50             }
51         }
52     }
53     //var_dump($rate);
54     ?>

```

ソーシャル情報を利用したシナリオ-先読みサーバ側 (create_scenario_w_social_proxy.php)

```
1 <?php
2 #argv[1]: 遅延情報ファイル,argv[2]: 書き込み先ファイル,$argv[3]: 最大個
   数,argv[4]: 乗車駅数制限
3 $file="./sep/".$argv[1];
4 $fp=fopen($file,"r");
5 $wfp=fopen($argv[2],"w");
6 $dsn = 'mysql:host=localhost;dbname=transit;charset=utf8';
7 $usr = 'yonemura';
8 $pass = 'ekripssoft';
9 $dbh = new PDO($dsn,$usr,$pass);
10
11 while(($data=fgetcsv($fp,",")) !== FALSE){
12     $line = $data[1];
13     $id = array();
14     $station = array();
15     $src = array();
16     $dsts = array();
17     $max = $argv[3];
18     try{
19         $sql = "select id from Line_memory where name= ?";
20         $stmt = $dbh->prepare($sql);
21         $stmt->execute(array($line));
22         while($result = $stmt->fetch(PDO::FETCH_ASSOC)){
23             $id[]=$result['id'];
24         }
25
26         #id から駅名を引っ張ってきて station へ入れる
27         #この数が爆発するので、制限する。まだ案はない
28         $sql = "select name from Station_memory where line_id= ? order by
   src_popularity desc limit ".$argv[4].";";
29         $stmt = $dbh->prepare($sql);
30         foreach($id as $line_id){
31             $stmt->execute(array($line_id));
32             while($result = $stmt->fetch(PDO::FETCH_ASSOC)){
33                 $station[] = $result['name'];
34             }
35         }
36         #station から WebSiteRanking,UserSearch を検索
37         $per_st=(int)$max/$argv[4];$lim=(int)$per_st/2;
38         $sql_wsr = "select id, dst from WebSiteRanking_memory where src = ?
   order by times desc, last_searched desc limit ".$lim.";";
39         $wsr_id = array();
40         $us_id = array();
41         $i=0;
42         $stmt = $dbh->prepare($sql_wsr);
```

```

43     foreach($station as $from){
44         $stmt->execute(array($from));
45         while($result = $stmt->fetch(PDO::FETCH_ASSOC)){
46             $src[] = $from;
47             $dsts[] = $result['dst'];
48             $wsr_id[] = $result['id'];
49             $i++;
50         }
51     }
52     $lim = (int)(($max-$i)/$argv[4]);
53     $sql_us = "select id, dst from UserSearch_memory where src = ? order
by times desc,last_searched desc limit ".$lim.";";
54     $wsr_id = array();
55     $stmt = $dbh->prepare($sql_us);
56     foreach($station as $from){
57         $stmt->execute(array($from));
58         while($result = $stmt->fetch(PDO::FETCH_ASSOC)){
59             $src[] = $from;
60             $dsts[] = $result['dst'];
61             $us_id[] = $result['id'];
62         }
63     }
64
65     #更新处理
66     $update_wsr = "update WebSiteRanking_memory set last_searched = now()
where id = ?;";
67     $update_us = "update UserSearch_memory set last_searched = now()
where id = ?;";
68
69     $stmt = $dbh->prepare($update_wsr);
70     foreach($wsr_id as $elm){
71         $stmt->execute(array($elm));
72     }
73     $stmt = $dbh->prepare($update_us);
74     foreach($us_id as $elm){
75         $stmt->execute(array($elm));
76     }
77 }catch(PDOException $e){
78     print($e->getMessage());
79 }
80 fwrite($wfp,"__,$data[0]."\n");
81 $count = count($src);
82 for($i=0;$i<$count;$i++){
83     fwrite($wfp,$src[$i] . "," . $dsts[$i]."\n");
84 }
85
86 }
87 ?>

```

システム性能評価用スクリプト-クライアント,先読みサーバ間 (client_to_proxy.php)

```
1 <?php
2 $st_num = $argv[1]; //駅名リストから駅を選択
3 $clients = $argv[2]; //同時接続クライアント数
4 $fp = fopen("stations","r");
5 $queries = array(); //投げるクエリ
6 $thread = array(); //スレッド保持する
7 $stations = array();
8 date_default_timezone_set('Asia/Tokyo');
9
10 /*作成するクエリに指定する時間*/
11 $yjYm = date("Ym");
12 $nYm = date("Y/n");
13
14 $yd = date("d");
15 $jnd = date("j");
16
17 $yh = date("H");
18 $jnh = date("G");
19
20 $m = date("i");
21 $yjm1 = substr($m,0,1);
22 $yjm2 = substr($m,1,1);
23 /*駅名を配列に*/
24 while(($str = fgets($fp)) !== false){
25     $stations[] = trim($str);
26 }
27 /*以下から post する内容作成*/
28 for($i = 0; $i < $clients; $i++){
29
30     $station = $stations[$st_num];
31     $rand = 1;
32     if($rand == 1){
33         $queries[] = "host=1&from=" . $station . "&to=&via1=&via2=&via3=&ym=" .
34 $yjYm . "&d=" . $yd . "&hh=" . $yh . "&m1=" . $yjm1 . "&m2=" . $yjm2 .
35 "&type=1&ticket=ic&al=1&shin=1&ex=1&hb=1&lb=1&sr=1&s=0&expkind=1&ws=2&kw=" .
36 $station;
37     }else if($rand == 2){
38         $queries[] = "host=2&from=" . $station . "&to=&via=&Dym=" . $yjYm .
39 "&Ddd=" . $jnd . "&Dhh=" . $jnh . "&Dmn1=" . $yjm1 . "&Dmn2=" . $yjm2 .
40 "&Cway=0&Cfp=1&C7=1&C2=0&C3=0&C1=0&C4=0&C6=2";
41     }else if($rand == 3){
42         $queries[] = "host=3&from=" . $station .
43 "&to=&via1=&via2=&via3=&month=" . $nYm . "&day=" . $jnd . "&hour=" . $jnh .
44 "&minute=" . $m .
45 "&basis=1&sort=0&wspeed=100&airplane=1&spreexprs=1&utrexprs=1&mtrplbus=1&othex
46 prs=1&intercitybus=1&ferry=1";
47     }
48 }
```

```

38     }
39 }
40 $u_code = 0;
41 $timestamp = microtime(true);
42 $timestamp += 3;
43 $i=0;
44 print("__init\n");
45 foreach($queries as $getquery){
46     $thread[] = new toPost($getquery,$u_code,$timestamp);
47     list($micro, $unixtime) = explode(" ", microtime());
48     $sec = $micro + date("s",$unixtime);
49     $time = date("Y/m/d H:i:",$unixtime) . $sec;
50     print("u_code: " . $u_code . ", time: " . $time . ", thread create\n");
51     $u_code++;$i++;
52 }
53 for($i=0; $i < $clients; $i++){
54     list($micro, $unixtime) = explode(" ", microtime());
55     $sec = $micro + date("s",$unixtime);
56     $time = date("Y/m/d H:i:",$unixtime) . $sec;
57     print("u_code: " . $i . ", time: " . $time . ", thread start\n");
58     $thread[$i] -> start();
59 }
60 print("__measurement\n");
61 foreach($thread as $th){
62     $th->join();
63 }
64 exit();
65
66 class toPost extends Thread{
67     private $query;
68     private $unique;
69     private $url
70     "http://192.168.225.190/pre_search_engine_test_make_query.php"; #query を生成す
71     だけの pre_search_engine
72     private $timestamp;
73     //private $station;
74     function __construct($q,$c,$t){
75         $this->query = $q;
76         $this->unique = $c;
77         $this->timestamp = $t;
78     }
79
80     public function run(){
81         $options = array("http" => array(
82             "method" => "POST",
83             "header" => "Content-Type: application/x-www-form-urlencoded",
84             "content" => $this->query,
85         ));

```



```

84     time_sleep_until($this->timestamp);
85     list($micro, $unixtime) = explode(" ", microtime());
86     $micro_float_b = microtime(true);
87     $sec_b = $micro + date("s",$unixtime);
88     $time = date("Y/m/d H:i",$unixtime) . $sec_b;
89     $str = $this->unique . "," . $time . "," . $unixtime . "," . $micro . ",";
90     $content = file_get_contents($this->url, false,
stream_context_create($options));
91     list($end, $micro_float_a) = explode("\n",$content);
92     $str .= $end;
93     $interval = $micro_float_a - $micro_float_b;
94     $str .= "," . $interval . "\n";
95     usleep($this->unique * 10000);
96     print($str);
97 }
98 }
99 }
100 ?>

```

システム性能評価用スクリプト-先読みサーバ,コンテンツ配信サーバ間(proxy_to_server.php)

```

1  <?php
2  $sessions = $argv[1];
3
4  $query = "http://192.168.225.191/testsession.php";
5  $hostname = "192.168.225.191"; //かわるかも
6
7  $ym = date("Ym");
8  $d = date("d");
9  $h = date("H");
10 $m = date("i");
11
12 $timestamp = time() + 4;
13
14 $u_code = 0;
15 $threads = array();
16
17 print("__init\n");
18 for($i = 0; $i < $sessions; $i++){
19 $attr = "?ym=" . $ym . "&d=" . $d . "&h=" . $h . "&m=" . $m . "&code=" . $u_code;
20 $threads[] = new GetData($u_code,$query.$attr,$hostname,$timestamp);
21 list($micro,$unixtime) = explode(" ",microtime());
22 $sec = $micro + date("s",$unixtime);
23 $time = date("Y/m/d H:i", $unixtime) . $sec;
24 print("u_code: " . $u_code . ", time: " . $time . "thread create\n");
25 $u_code++;
26 }

```

```

27
28 for($i = 0;$i < $sessions; $i++){
29     list($micro,$unixtime) = explode(" ",microtime());
30     $sec = $micro + date("s",$unixtime);
31     $time = date("Y/m/d H:i", $unixtime) . $sec;
32     print("u_code: " . $i . ", time: " . $time . "thread start\n");
33     $threads[$i]->start();
34 }
35 print("__measurement\n");
36 foreach($threads as $th){
37     $th->join();
38 }
39 exit;
40 class GetData extends Thread{
41     private $host;
42     private $query;
43     private $unique;
44     private $timestamp;
45     function __construct($u,$q,$h,$t){
46         $this->host = $h;
47         $this->query = $q;
48         $this->unique = $u;
49         $this->timestamp = $t;
50     }
51
52     public function run(){
53         time_sleep_until($this->timestamp);
54         list($micro,$unixtime) = explode(" ",microtime());
55         $micro_float_b = microtime(true);
56         $sec_b = $micro + date("s",$unixtime);
57         $time = date("Y/m/d H:i:", $unixtime) . $sec_b;
58         $str = $this->unique . "," . $time . "," . $unixtime . "," . $micro .
",,";
59         $options = array(
60             "http" => array(
61                 "method" => "GET",
62                 "proxy" => "tcp://localhost:8080",
63                 "request_fulluri" => true,
64                 "header" => "Content-Type:
application/x-www-form-urlencoded",
65             ),
66         );
67         $content = file_get_contents($this->query, false,
stream_context_create($options));
68         //$str .= $svr_return . ",";
69         list($micro,$unixtime) = explode(" ",microtime());
70         $micro_float_a = microtime(true);
71         $sec_a = $micro + date("s",$unixtime);

```

```

72     $time = date("Y/m/d H:i:", $unixtime) . $sec_a;
73
74     $svr_return = explode("<->", $content);
75     $str .= $svr_return[0] . "," . $time . "," . $unixtime . "," . $micro;
76     $interval = $micro_float_a - $micro_float_b;
77     $str .= "," . $interval . "\n";
78     usleep($this->unique * 10000);
79     print($str);
80
81     }
82 }
83 ?>

```

応答性能評価-実サービス提供者サーバ用スクリプト(hitstest.php)

```

1 <?php
2 $fp = fopen($argv[2], "r"); //stations src dst list
3 $src = array();
4 $dst = array();
5 $threads = array();
6 /*insert src dst list in array*/
7 while(($data = fgetcsv($fp, 1000, ",", "")) !== FALSE){
8     $src[] = $data[0];
9     $dst[] = $data[1];
10 }
11 /**/
12 $count = count($src);
13 $timestamp = microtime(true);
14 $now_unixtime = time();
15 print("__init\n");
16 for($i = 0; $i < $argv[1]; $i++){
17     $host = 1; // $argv[3]; //rand(1,3);
18     $st_num = 10; //rand(0, ($count-1));
19     $delay_mergin = 5; //rand(1,5);
20     if($host == 1){ /*yahoo transit*/
21         $ym = date("Ym");
22         $d = date("d");
23         $h = date("H");
24         $m = date("i");
25         $m1 = substr($m, 0, 1);
26         $m2 = substr($m, 1, 1);
27         /*queries*/
28         $post_query = "host=1&from=" . urlencode($src[$i]) .
"&to=&via1=&via2=&via3=&ym=" . $ym . "&d=" . $d . "&hh=" . $h . "&m1=" . $m1 .
"&m2=" . $m2 .
"&type=1&ticket=ic&al=1&shin=1&ex=1&hb=1&lb=1&sr=1&s=0&expkind=1&ws=2&kw=";

```

```

29 $get_query="http://192.168.225.190/redirecttest.php?host=1&from=" .
urlencode($src[$i]) . "&to=" . urlencode($dst[$i]) .
"&via1=&via2=&via3=&ym=" . $ym . "&d=" . $d . "&hh=" . $h . "&m1=" . $m1 . "&m2=" .
$m2 .
"&type=1&ticket=ic&al=1&shin=1&ex=1&hb=1&lb=1&sr=1&s=0&expkind=1&ws=2&kw=" .
urlencode($dst[$i]);
30 /*create thread*/
31 /*setting delay time*/
32 $delay = $timestamp + $delay_mergin;
33 $threads[] = new
SendData($post_query,$get_query,$src[$i],$dst[$i],$i,$host,$delay);

34
35 /*time of creating thread*/
36 list($micro, $unixtime) = explode(" ", microtime());
37 $sec = $micro + date("s",$unixtime);
38 $time = date("Y/m/d H:i:",$unixtime) . $sec;
39 print("u_code: " . $i . ", time: " . $time . ", thread create\n");
40
41 }else if($host == 2){
42     $ym = date("Ym");
43     $d = date("j");
44     $h = date("G");
45     $m = date("i");
46     $m1 = substr($m,0,1);
47     $m2 = substr($m,1,1);
48     /*queries*/
49     $post_query = "host=2&from=" . urlencode($src[$i]) .
"&to=&via=&Dym=" . $ym . "&Ddd=" . $d . "&Dhh=" . $h . "&Dmn1=" . $m1 . "&Dmn2=" .
$m2 . "&Cway=0&Cfp=1&C7=1&C2=0&C3=0&C1=0&C4=0&C6=2";

50     $get_query =
"http://192.168.225.190/redirecttest.php?host=2&eki1=" .
urlencode($src[$i]) . "&eki2=" . urlencode($dst[$i]) . "&eki3=&via_on=1&Dym=" .
$ym . "&Ddd=" . $d . "&Dhh=" . $h . "&Dmn1=" . $m1 . "&Dmn2=" . $m2 .
"&Cway=0&Cfp=1&C7=1&C2=0&C3=0&C1=0&C4=0&C6=2&S=%E6%A4%9C%E7%B4%A2&Cmap1=&rf
=nr&pg=0&eok1=&eok2=&eok3=&Csg=1";

51     /*create thread*/
52     /*setting delay time*/
53     $delay = $timestamp + $delay_mergin;
54     $threads[] = new
SendData($post_query,$get_query,$src[$i],$dst[$i],$i,$host,$delay);

55
56     /*time of creating thread*/
57     list($micro, $unixtime) = explode(" ", microtime());
58     $sec = $micro + date("s",$unixtime);
59     $time = date("Y/m/d H:i:",$unixtime) . $sec;
60     print("u_code: " . $i . ", time: " . $time . ", thread create\n");
61
62 }else{
63     $ym = date("Y/m");

```

```

64         $d = date("j");
65         $h = date("G");
66         $m = date("i");
67
68         /*queries*/
69         $post_query = "host=3&from=" . urlencode($src[$i]) .
"&to=&via1=&via2=&via3=&month=" . $ym . "&day=" . $d . "&hour=" . $h .
"&minute=" . $m .
"&basis=1&sort=0&wspeed=100&airplane=1&sprexprs=1&utrexprs=1&mtrplbus=1&oth
exprs=1&intercitybus=1&ferry=1";
70         $get_query =
"http://192.168.225.190/redirecttest.php?host=1&orvStationName=" .
urlencode($src[$i]) . "&dnvStationName=" . urlencode($dst[$i]) . "&month=" .
$ym . "&day=" . $d . "&hour=" . $h . "&minute=" . $m .
"&sort=0&wspeed=100&airplane=1&sprexprs=1&utrexprs=1&mtrplbus=1&othexprs=1&
intercitybus=1&ferry=1&orvStationCode=&dnvStationCode=&basis=1&ctl=020010&a
tr=2&detailset=1&init=";
71         /*create thread*/
72         /*setting delay time*/
73         $delay = $timestamp + $delay_mergin;
74         $threads[] = new
SendData($post_query,$get_query,$src[$i],$dst[$i],$i,$host,$delay);
75
76         /*time of creating thread*/
77         list($micro, $unixtime) = explode(" ", microtime());
78         $sec = $micro + date("s",$unixtime);
79         $time = date("Y/m/d H:i:",$unixtime) . $sec;
80         print("u_code: " . $i . ", time: " . $time . ", thread create\n");
81
82     }
83 }
84 /*thread start*/
85 for($i =0; $i < $argv[1]; $i++){
86     list($micro, $unixtime) = explode(" ", microtime());
87     $sec = $micro + date("s",$unixtime);
88     $time = date("Y/m/d H:i:",$unixtime) . $sec;
89     print("u_code: " . $i . ", time: " . $time . ", thread start\n");
90     $threads[$i]->start();
91     //sleep(5);
92 }
93
94 print("__measurement\n");
95 /*thread join*/
96 foreach($threads as $thread){
97     $thread->join();
98 }
99
100 exit();
101
102 class SendData extends Thread{

```

```

103     private $pre_query;
104     private $post_query;
105     private $src;
106     private $dst;
107     private $unique;
108     private $host;
109     private $engine = "http://192.168.225.190/pre_search_engine.php";
110     private $timestamp;
111
112     function __construct($preq,$postq,$s,$d,$u,$h,$t){
113         $this->pre_query = $preq;
114         $this->post_query = $postq;
115         $this->src = $s;
116         $this->dst = $d;
117         $this->unique = $u;
118         $this->host = $h;
119         $this->timestamp = $t;
120     }
121
122     public function run(){
123         /*configure options*/
124         /*for pre_search_engine*/
125         $post_options = array("http" => array(
126             "method" => "POST",
127             "header" => "Content-Type: application/x-www-form-urlencoded",
128
129             "content" => $this->pre_query,
130         ));
131         /*for origin*/
132         $get_header = array(
133             "Content-Type: application/x-www-form-urlencoded",
134
135             "Accept: /*/*",
136             "User-Agent: Mozilla/5.0",
137         );
138         $get_options = array(
139             "http" => array(
140                 "method" => "GET",
141                 "header" => implode("\r\n",$get_header),
142                 "proxy" => "tcp://192.168.225.190:8080",
143                 "request_fulluri" => true,
144             ),
145         );
146
147         /*send post to pre_search engine*/
148         time_sleep_until($this->timestamp);
149         list($micro, $unixtime) = explode(" ", microtime());
150         $micro_float_before_post = microtime(true);

```

```

149     $sec_before_post = $micro + date("s",$unixtime);
150     $time = date("Y/m/d H:i:",$unixtime) . $sec_before_post;
151     $str = $this->unique . "," . $this->host . "," . $this->src . "," .
$this->dst . "," . $time . "," . $unixtime . "," . $micro . ",";
152
153     $content = file_get_contents($this->engine, false,
stream_context_create($post_options));
154
155     list($micro, $unixtime) = explode(" ", microtime());
156     $micro_float_after_post = microtime(true);
157     $sec_after_post = $micro + date("s",$unixtime);
158     $time = date("Y/m/d H:i:",$unixtime) . $sec_after_post;
159     $delay_post = $micro_float_after_post - $micro_float_before_post;
160     $str .= $time . "," . $unixtime . "," . $micro . "," . $delay_post .
",";
161     //print($content);
162     sleep(3);
163     /*send get to origin server*/
164     /*if cache hits, content return from proxy*/
165     list($micro, $unixtime) = explode(" ", microtime());
166     $micro_float_before_get = microtime(true);
167     $sec_before_get = $micro + date("s",$unixtime);
168     $time = date("Y/m/d H:i:",$unixtime) . $sec_before_get;
169     $str .= $time . "," . $unixtime . "," . $micro . ",";
170
171     file_get_contents($this->post_query,false,stream_context_create($get_option
s));
172     //print($content);
173     list($micro, $unixtime) = explode(" ", microtime());
174     $micro_float_after_get = microtime(true);
175     $sec_after_get = $micro + date("s",$unixtime);
176     $time = date("Y/m/d H:i:",$unixtime) . $sec_after_get;
177     $delay_get = $micro_float_after_get - $micro_float_before_get;
178     $str .= $time . "," . $unixtime . "," . $micro . "," . $delay_get . "," .
$content . "\n";
179
180     print($str);
181
182 }
183 }
184
185 ?>

```

応答性能評価-実験用コンテンツ配信サーバ用スクリプト (whole_test.php)

```

1 <?php
2 /*inicialize stations*/
3 $hour = date("G");

```

```

4 $file = "./make_stations/scenario_".$hour;
5 //$file = "./make_stations/scenario_ranking";
6 $fp = fopen($file,"r");
7 /*シナリオは一時間ごと別ファイル*/
8 /*スクリプトも一時間ごと再起動*/
9
10 if(($hour >= 2) && ($hour <= 5)){return;}
11 $times_ary
array(10,10,0,0,0,0,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10); =
12 $times_per_sec = $times_ary[$hour];
13 $unique = 0;
14 $thread_ary = array();
15 $th_ary_num=0;
16 $minute_now = (int)date("i");
17 $second_now = (int)date("s");
18 $sixty_minutes = 3600 - (60*$minute_now) - $second_now;
19 for($i=0;$i<$sixty_minutes;$i++){
20     /*駅リスト作成*/
21     $ary = array();$times=0;
22     for($j=0;$j<$times_per_sec;$j++){
23         $line = fgets($fp);
24         $line = str_replace(array("\r\n","\r","\n"),"", $line);
25         $ary[] = $line;
26         $times++;
27     }
28     $push = mt_rand(0,1);
29     if($push == 0){
30         $timestamp = time();
31     }else{
32         $timestamp = microtime(true);
33     }
34
35     $thread_ary[$th_ary_num] = new
CreateSendThread($ary,$times,$unique,$timestamp,$push);
36     $thread_ary[$th_ary_num]->start();
37     $th_ary_num++;if($th_ary_num>=300){$th_ary_num=0;}
38     $unique += $times;
39     sleep(1);
40 }
41
42
43
44 class CreateSendThread extends Thread{
45     private $station_ary;
46     private $times;
47     private $unique_start;
48     private $timestamp;
49     private $type;

```



```

50
51     function __construct($ary,$t,$u_start,$t,$p){
52         $this->station_ary = $ary;
53         $this->times = $t-1;
54         $this->unique_start = $u_start;
55         $this->timestamp = $t;
56         $this->type = $p;
57     }
58
59     public function run(){
60         $threads = array();
61         $delay_mergin = 6;
62         $print_delay = $this->timestamp + $delay_mergin;
63         foreach($this->station_ary as $srcdst){
64             $srcdst_ary = explode(",",$srcdst);
65             $host = 1; //rand(1,3);
66
67             if($host == 1){ /*yahoo transit*/
68                 $ym = date("Ym");
69                 $d = date("d");
70                 $h = date("H");
71                 $m = date("i");
72                 $m1 = substr($m,0,1);
73                 $m2 = substr($m,1,1);
74                 /*queries*/
75                 $post_query = "host=1&from=" . urlencode($srcdst_ary[0]) .
"&to=&via1=&via2=&via3=&ym=" . $ym . "&d=" . $d . "&hh=" . $h . "&m1=" . $m1 .
"&m2=" . $m2 .
"&type=1&ticket=ic&al=1&shin=1&ex=1&hb=1&lb=1&sr=1&s=0&expkind=1&ws=2&kw=";
76
$get_query="http://192.168.225.190/redirecttest_inner.php?host=1&from=" .
urlencode($srcdst_ary[0]) . "&to=" . urlencode($srcdst_ary[1]) .
"&via1=&via2=&via3=&ym=" . $ym . "&d=" . $d . "&hh=" . $h . "&m1=" . $m1 . "&m2=" .
$m2 .
"&type=1&ticket=ic&al=1&shin=1&ex=1&hb=1&lb=1&sr=1&s=0&expkind=1&ws=2&kw=" .
urlencode($srcdst_ary[1]);
77                 /*create thread*/
78                 /*setting delay time*/
79                 $delay = $this->timestamp + $delay_mergin;
80                 $threads[] = new
SendData($post_query,$get_query,$srcdst_ary[0],$srcdst_ary[1],$this->unique_
start,$host,$delay,$this->type);
81
82                 }else if($host == 2){
83                     $ym = date("Ym");
84                     $d = date("j");
85                     $h = date("G");
86                     $m = date("i");
87                     $m1 = substr($m,0,1);
88                     $m2 = substr($m,1,1);

```

```

89         /*queries*/
90         $post_query = "host=2&from=" . urlencode($srcdst_ary[0]) .
"&to=&via=&Dym=" . $ym . "&Ddd=" . $d . "&Dhh=" . $h . "&Dmn1=" . $m1 . "&Dmn2=" .
$m2 . "&Cway=0&Cfp=1&C7=1&C2=0&C3=0&C1=0&C4=0&C6=2";
91         $get_query =
"http://192.168.225.190/redirecttest_inner.php?host=2&eki1="
urlencode($srcdst_ary[0]) . "&eki2=" . urlencode($srcdst_ary[1]) .
"&eki3=&via_on=1&Dym=" . $ym . "&Ddd=" . $d . "&Dhh=" . $h . "&Dmn1=" . $m1 .
"&Dmn2=" . $m2
"&Cway=0&Cfp=1&C7=1&C2=0&C3=0&C1=0&C4=0&C6=2&S=%E6%A4%9C%E7%B4%A2&Cmap1=&rfrf=
nr&pg=0&eok1=&eok2=&eok3=&Csg=1";
92         /*create thread*/
93         /*setting delay time*/
94         $delay = $this->timestamp + $delay_mergin;
95         $threads[] = new
SendData($post_query,$get_query,$srcdst_ary[0],$srcdst_ary[1],$this->unique_
start,$host,$delay,$this->type);
96
97
98     }else{
99         $ym = date("Y/m");
100        $d = date("j");
101        $h = date("G");
102        $m = date("i");
103
104        /*queries*/
105        $post_query = "host=3&from=" . urlencode($srcdst_ary[0]) .
"&to=&via1=&via2=&via3=&month=" . $ym . "&day=" . $d . "&hour=" . $h . "&minute=" .
$m
"&basis=1&sort=0&wspeed=100&airplane=1&sprexprs=1&utrexprs=1&mtrplbus=1&othe
xprs=1&intercitybus=1&ferry=1";
106        $get_query =
"http://192.168.225.190/redirecttest_inner.php?host=1&orvStationName="
urlencode($srcdst_ary[0]) . "&dnvStationName=" . urlencode($srcdst_ary[1]) .
"&month=" . $ym . "&day=" . $d . "&hour=" . $h . "&minute=" . $m .
"&sort=0&wspeed=100&airplane=1&sprexprs=1&utrexprs=1&mtrplbus=1&othexprs=1&i
ntercitybus=1&ferry=1&orvStationCode=&dnvStationCode=&basis=1&ctl=020010&atr
=2&detailset=1&init=";
107        /*create thread*/
108        /*setting delay time*/
109        $delay = $this->timestamp + $delay_mergin;
110        $threads[] = new
SendData($post_query,$get_query,$srcdst_ary[0],$srcdst_ary[1],$this->unique_
start,$host,$delay,$this->type);
111    }
112    $this->unique_start += 1;
113
114    }
115    foreach($threads as $th){
116        $th->start();
117    }
118    foreach($threads as $th){

```

```

119         $th->join();
120     }
121 }
122 }
123
124 class SendData extends Thread{
125     private $pre_query;
126     private $post_query;
127     private $src;
128     private $dst;
129     private $unique;
130     private $host;
131     private $engine
132     "http://192.168.225.190/pre_search_engine_for_exp_inner.php";
133     private $timestamp;
134     private $type;
135     function __construct($preq,$postq,$s,$d,$u,$h,$t,$ty){
136         $this->pre_query = $preq;
137         $this->post_query = $postq;
138         $this->src = $s;
139         $this->dst = $d;
140         $this->unique = $u;
141         $this->host = $h;
142         $this->timestamp = $t;
143         $this->type = $ty;
144     }
145     public function run(){
146         /*configure options*/
147         /*for pre_search_engine*/
148         $post_options = array("http" => array(
149             "method" => "POST",
150             "header" => "Content-Type: application/x-www-form-urlencoded",
151
152             "content" => $this->pre_query,
153         ));
154         /*for origin*/
155         $get_header = array(
156             "Content-Type: application/x-www-form-urlencoded",
157
158             "Accept: /*/*",
159             "User-Agent: Mozilla/5.0",
160         );
161         $get_options = array(
162             "http" => array(
163                 "method" => "GET",
164                 "header" => implode("\r\n",$get_header),
165                 "proxy" => "tcp://192.168.225.190:8080",
166                 "request_fulluri" => true,

```

```

165         ),
166     );
167
168     /*send post to pre_search engine*/
169     if($this->timestamp < time()){
170         sleep(3);
171     }else{
172         time_sleep_until($this->timestamp);
173     }
174     list($micro, $unixtime) = explode(" ", microtime());
175     $micro_float_before_post = microtime(true);
176     $sec_before_post = $micro + date("s",$unixtime);
177     $time = date("Y/m/d H:i:",$unixtime) . $sec_before_post;
178
179     $str = $this->unique . "," . $this->type . "," . $this->host . "," .
180     $this->src . "," . $this->dst . "," . $time . "," . $unixtime . "," . $micro .
181     ",";
182
183     $content = file_get_contents($this->engine, false,
184     stream_context_create($post_options));
185
186     list($micro, $unixtime) = explode(" ", microtime());
187     $micro_float_after_post = microtime(true);
188     $sec_after_post = $micro + date("s",$unixtime);
189     $time = date("Y/m/d H:i:",$unixtime) . $sec_after_post;
190     $delay_post = $micro_float_after_post - $micro_float_before_post;
191
192     $str .= $time . "," . $unixtime . "," . $micro . "," . $delay_post . ",";
193
194     sleep(3);
195     /*send get to origin server*/
196     /*if cache hits, content return from proxy*/
197     list($micro, $unixtime) = explode(" ", microtime());
198     $micro_float_before_get = microtime(true);
199     $sec_before_get = $micro + date("s",$unixtime);
200     $time = date("Y/m/d H:i:",$unixtime) . $sec_before_get;
201     $str .= $time . "," . $unixtime . "," . $micro . ",";
202
203     file_get_contents($this->post_query,false,stream_context_create($get_options
204     ));
205     list($micro, $unixtime) = explode(" ", microtime());
206     $micro_float_after_get = microtime(true);
207     $sec_after_get = $micro + date("s",$unixtime);
208     $time = date("Y/m/d H:i:",$unixtime) . $sec_after_get;
209     $delay_get = $micro_float_after_get - $micro_float_before_get;
210
211     $str .= $time . "," . $unixtime . "," . $micro . "," . $delay_get . "," .
212     $content."\n";
213
214     print($str);
215 }

```

206	}
207	?>