

位置情報を制約とした Web の事前 Cache 手法

佐々木 威 河川 信夫

名古屋大学大学院 工学研究科

Web PreCaching Method based on Location Information

Takeshi Sasaki Nobuo Kawaguchi

Graduate school of Engineering, Nagoya University

E-mail: sasaki@ucl.nuee.nagoya-u.ac.jp, kawaguti@nuee.nagoya-u.ac.jp

概要 近年,“いつでも,どこでも”ネットワークに接続し,オンライン上の情報を検索・利用したいという要望が高まってきている.しかし,究極的に“いつでも,どこでも”ネットワークに接続できる環境を実現する事は難しい.よって,本稿では,ネットワークに接続していなくても,事前にユーザが使いそうな Web サイトやアプリケーションから情報を集め保存しておき,オフライン時に保存された情報を引き出すことによって,ネットワークへ接続できなくとも Web 上の情報・アプリケーションを利用できる仕組みを提案・実装し,評価をする.

1.はじめに

近年,日本ではインターネット利用者数が増加の一途をたどっている.また,総務省の総務省通信白書[1]によると,携帯移動端末において,インターネットの利用者数が増加していることが分かる[図1].

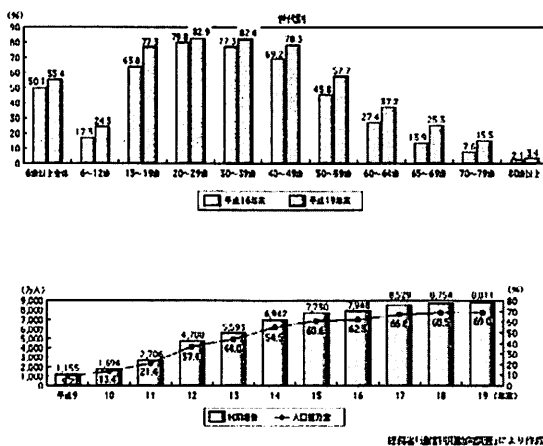


図1:携帯電話におけるインターネット利用者の増加(上)

インターネット利用者数の推移 (下)

このことから,ユーザが“いつでも,どこでも”インターネットに接続し利用するという傾向が高まってきていると言える.しかしながら,本当の意味で“いつでも,どこでも”インターネットに接続できるよう環境が整う未来は遠いのが現状である.これは,地下鉄やトンネル,時には航空機内など,ネットワークへの接続が難しい様々な場面があるからである.このような場面でオンライン上の情報を検索することは難しい.だが一方で,ネットワークに接続できない時,ユーザはどのような情報を調べようとして

いたのか疑問が生じる.ネットワークへ接続できなくなる状況というのは,ある程度ユーザにも予測がつくからだ.それは,ネットワークへ接続できなくなる状況が先程述べたような,地下鉄やトンネル等のネットワークへの接続が難しくなるとユーザにも一目で分かるような状況だからだ.それに関わらず,ユーザはネットワークへ接続し Web を閲覧して情報を得ようとした.この状況からユーザは早急に調べなくてはいけない情報を検索しようとした可能性が高いと推測できる.では,早急に調べなくてはいけない情報とは一体何だろうか.我々はそれがユーザの位置と移動に関わる情報だったのではないかと考えた.誰も,初めて行く土地で道が分からなくなり,自分の現在地をネットワークへ接続して調べたことがあるのではないだろうか,出張へ向かう電車の中で乗り継ぎ電車の時間を調べたことがあるのではないだろうか.その時はたまたまネットワークに接続できたのかもしれない.もしくは事前に調べた情報で十分だったのかもしれない.ネットワークから情報を検索できなくとも運良く問題を乗り越えられたのかもしれない.しかし,それがネットワークへ接続できない環境だったらどうなっていたのだろうか.また,事前に調べるにしても手間がかかりはしなかっただろうか.運悪く,困難を乗り越えられなかったらどうなっていただろうか.現状のネットワーク環境では“いつでも,どこでも”ネットワークに接続できるわけではなく,ネットワークに接続できない瞬間に様々な困難が潜んでいると言っても過言ではない.そこで本稿では,これらの問題を解決し困難を回避するために,ユーザがネットワーク接続可能時に前もって様々な Web 上の情報をキャッシュしておく PreCache を提案する.ここで言うキャッシュとは Web Proxy や Web ブラウザが保持する Web キャッシュの事である.これらのキャッシュはユーザが Web ページを閲覧する際に初めて作成されるものである.しか

し、我々が提案する PreCache はオフライン時にユーザが必要としそうな Web の情報を保存する点で差異がある。また、Web キャッシュはユーザが自分で Web ページを閲覧しない限り作成できないので、ユーザが全て自分の手で Web キャッシュを作成するには膨大な手間がかかる。よって、PreCache はユーザに負荷を与えないように自動で収集される必要がある。

しかし、単純に PreCache を作ると言っても容易ではない。PreCache は前もって Web の情報を収集し保存する手法なので、情報を収集する時点でオフライン時にユーザがどんな情報を欲しがるか、ある程度絞り込めていなければならない。しかし、情報を絞り込むためには何らかのキーが必要になるのは明白である。そこで我々は何が情報を絞り込むためのキーになるかを模索し、位置情報が重要なキーとなるのではないだろうかと考えに至った。位置情報をキーとすれば、ユーザが宿泊するホテルの位置や電車の乗換えをしなければならない駅の位置、ユーザが移動した先の位置周辺に存在するレストランの情報など、ユーザが突発的に調べたいと思う情報がある程度 Web の情報から絞り込むことができるからである。よって、本研究ではユーザの位置に関する情報を制約とし、Web から情報を絞り込み、事前に取得・保存する PreCache を提案する。

本稿では第 2 章において位置情報を制約として PreCache を作成するシステムの設計について述べ、第 3 章ではプロトタイプの実装について触れ、第 4 章においてプロトタイプを用いて作成した PreCache システムについて評価・考察し、第 5 章において本研究のまとめを述べる。

2. システム設計

本章では、PreCache を作成するためのシステムの要件とシステム設計について記す。

2-1. システム要件

PreCache を作る為のシステム要件として、以下の事項が挙げられる。

2-1-1. 位置情報の取得

PreCache を作成するに当たって、位置情報がユーザの状況を推測するのに重要なキーとなり得ることは前節で述べた。よって、ユーザの位置に関する情報を取得することが要件として挙がる。本稿では、ユーザの現在地や目的地などの位置情報をユーザの入力により取得する。また、キーとなる位置情報はユーザの入力した情報だけでなく、その経路や移動にも存在する。よって、ユーザの入力から、経路探索などを行い入力され位置情報以外の情報を取得することも要件として挙がる。

2-1-2. 位置情報を制約とした Web からの情報収集

位置情報を使用することは、PreCache を作成するための重要な足掛かりとなる可能性がある。しかし、位置情報を制約として Web から情報を収集すると言っても、単純に

位置情報をキーにした検索を行って、キーとマッチした Web の情報をキャッシュすればいいと言うわけではない。何故ならば、今日のインターネットにはアプリケーションや画像など様々な種類の情報があり、アプリケーションの実行結果、またはそのアプリケーションごとキャッシュをしたり、画像情報なども全てキャッシュしなければ、Web をキャッシュするとは言い難いからである。よって、様々なアプリケーション・Web サイト・ファイルタイプに対応し情報を抽出してくるモジュール群が必要となる。

2-1-3. 収集した情報の整形と保存

Web ページや Web アプリケーション等から情報をキャッシュするとしても、キャッシュをユーザのオフライン時に利用できる形で保存しなければならない。Web Proxy や Web ブラウザの Web キャッシュの場合は、ただ単純にユーザが閲覧した Web 情報を保存しただけであり、キャッシュを検索・利用しようとする際に困難が生じる可能性が高いと考えられる。よって、オフライン時にユーザが PreCache を利用して必要な情報を検索、容易に利用できるように情報を整形し、保存する必要がある。本研究では、作成した PreCache とその PreCache を作成した位置情報のキーに関連を持たせた上で保存することで、PreCache の検索を可能にする。また、PreCache とキーとを関連付けたまま保存するシステムとしてリレーショナルデータベースが最適だと考え、リレーショナルデータベースにデータを保存する方針とする。

2-1-4. 携帯移動端末への対応

PreCache は主にユーザが移動中もしくは移動先などでネットワークに接続不可能な状況で利用する状況を想定している。よって、移動中もしくは移動先、かつネットワーク接続不可能という環境を想定して PreCache を利用可能にする必要がある。そして移動中に使える端末はラップトップや PDA、携帯電話等の携帯移動端末に限られる。よって、PreCache はこれらの携帯移動端末で利用可能なものとする必要がある。また、PreCache を作成する工程で、Web をクロールし集めてきた情報を PreCache として保存する処理は携帯移動端末にとって負荷が大き過ぎる。よって、PreCache を作るサーバをネットワーク上に設置し、端末はクライアントとしてサーバに対してアクセスしユーザから入力された条件を送る。サーバはその条件にあった Web の情報を PreCache として保存し、端末へ送り返すことによって、端末側の負荷を小さくする。また、サーバ側も一度集めた PreCache をサーバに保存しておくことにより、クライアントから同じ要求が来ても再度 PreCache を作成する手間を省き、負荷を軽減できる。

2-1-5. 不足情報の再取得

事前に Web の情報をキャッシュするといっても、ユーザに必要な情報がすべて PreCache に収まっているとは限らない。そこで、不足した情報を再度取得する為にサーバへ問合せをしなくてはならない。しかし、前にも述べたよ

うに PreCache を使用する環境はオフライン状態にあることを示している。示している可能性が高いので、すぐにサーバへ問合せをすることができないと考えられる。これを解決するために、不足した情報を取得するためのクエリをユーザ端末に一時的に保持しておき、ホットスポットなどでネットワーク接続可能になった際に改めて不足した PreCache を作成する手法をとることによって、不足した情報も取得できるようにする。

2-2. システム設計

本章では、システムの設計を示すと共に、システムの構成要素について述べる。システム要件を踏まえた設計を図 2 に示す。

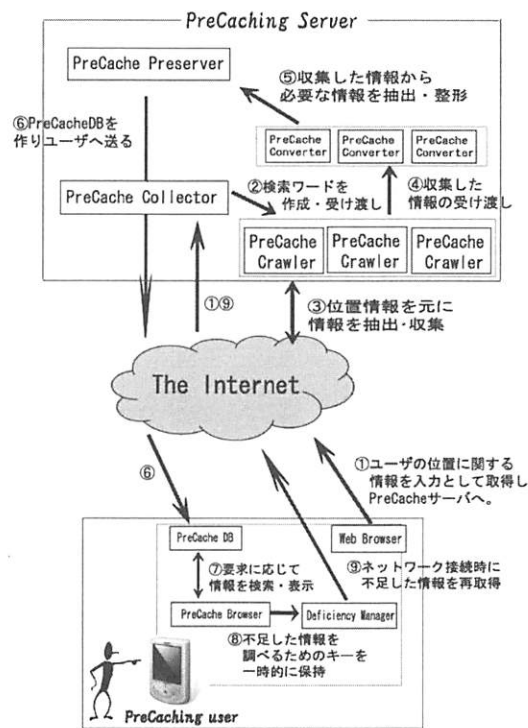


図 2: PreCache システム設計図

2-2-1. PreCache Collector

PreCache Collector はユーザから受け取った入力（現在地、目的地）を元に、経路探索を行い現在地から目的地までの駅名や街名をリストアップする。リストアップされた駅名・街名などの位置情報は Web に検索をかけるキーとして、PreCache Crawler へ送られる。

2-2-2. PreCache Crawler

PreCache Crawler は PreCache Collector から受け取った駅名・街名のリストを使い、Web から情報を収集する。しかし、PreCache Crawler は一つだけではない。これは Web 上に様々なタイプの情報が存在するため、各タイプの情報に特化した Crawler が必要だからである。例として、Web 上の html ドキュメントを集めるために、Google 検索で検索をし、条件に当てはまった Web ページの情報を取得してくるなどの機能が挙げられる。また、PreCache

Crawler は各々の特化した情報を集め、各タイプの情報を解析するのに最適化された PreCache Converter へ送る。

2-2-3. PreCache Converter

前項でも述べたように、PreCache Converter は PreCache Crawler が集めてきた情報を解析し、整形する。例えば、PreCache Crawler がぐるなび[2]から検索条件に沿ったレストラン情報を xml 形式で取得し PreCache Converter へ送るといった機能が挙げられる。ぐるなびは株式会社ぐるなびが運営する飲食店の情報を集めたウェブサイトである。飲食店の情報を飲食店事業主から広告として募り、飲食店が各店舗の情報を登録・情報発信し利用者は無料で飲食店の情報を検索・閲覧できるサイトである。そして、PreCache Converter は受け取った xml を解析し整形した上で、PreCache Preserver へ送る。また、PreCache Crawler の例を図 3 に PreCache Converter の例を図 4 にそれぞれ示す。

```
<?xml version="1.0" encoding="UTF-8" ?>
<rest order="0">
  <id>n616400</id>
  <update_date>2008-08-17 03:15:38</update_date>
  <name>鉄板焼 お好み焼 周〜JYU〜</name>
  <name_kana>テッパンオホコノミヤキジュウ</name_kana>
  <latitude>35.169697</latitude>
  <longitude>136.908458</longitude>
  <category>鉄板焼</category>
  <url>http://r.gnavi.co.jp/n616400/?ak=MVlrm81pT7f0LAt1rhBuiYRCnHafQc</url>
  <url_mobile>http://mobile.gnavi.co.jp/n616400/?ak=MVlrm81pT7f0LAt1rhB</url_mobile>
  <image_url>
    <shop_image1>http://apicache.gnavi.co.jp/image/rest/index.php?img=n6</shop_image1>
    <shop_image2 />
    <groode>http://apicache.gnavi.co.jp/image/rest/index.php?img=n616400</groode>
  </image_url>
  <address>〒460-0003 愛知県名古屋市中区錦3-4</address>
  <tel>052-955-3800</tel>
  <fax />
  <open_time>月〜金 17:00〜03:00(L.O.02:00)<BR>土 17:00〜24:00(L.O.23:00)</open_time>
  <holiday>日・祝</holiday>
  <access>
    <line>地下鉄名城・名港線</line>
    <station>久屋大通駅</station>
    <station_exit>4番出口</station_exit>
    <walk>2</walk>
    <note />
  </access>
</rest>
```

図 3 : ぐるなび PreCache Crawler

(ぐるなびから xml 形式でレストラン情報を取得する)

```
-----
店名: 鉄板焼 お好み焼 周〜JYU〜
緯度: 35.169697
経度: 136.908458
URL: http://r.gnavi.co.jp/n616400/?ak=MVlrm81pT7f0LAt1rhBuiYRCnHafQc&stno=F32BF28330
電話番号: 052-955-3800
営業時間: 月〜金 17:00〜03:00(L.O.02:00)<BR>土 17:00〜24:00(L.O.23:00)
休日: 日・祝
路線名: 地下鉄名城・名港線
最寄駅: 久屋大通駅
県名: 愛知県
種別: 和食
-----
店名: さんまろ 徳吉
緯度: 35.169444
経度: 136.907778
URL: http://r.gnavi.co.jp/n624703/?ak=MVlrm81pT7f0LAt1rhBuiYRCnHafQc&stno=F32BF28330
店舗: http://apicache.gnavi.co.jp/image/rest/index.php?img=n624703
画像: http://apicache.gnavi.co.jp/image/rest/index.php?img=n624703
電話番号: 052-971-5530
営業時間: 17:00〜01:00(L.O.02:00)<BR>日・祝 17:00〜24:00(L.O.23:00)
休日: 無休
路線名: 地下鉄名城
最寄駅: 久屋大通駅
県名: 愛知県
種別: 和風居酒屋
-----
```

図 4 : ぐるなび PreCache Converter の出力結果例

2-2-4. PreCache Preserver

PreCache Preserver は PreCache Converter から受け取

った整形された情報をリレーショナルデータベース（以降、情報を格納したデータベースを PreCacheDB と呼ぶ）へ保存し、PreCacheDB をユーザへ送る。PreCache Preserver は更に、一度格納された PreCacheDB を一定期間保持し、PreCache Server へ過去にあった要求があった場合に再び PreCache を作ることなく、ユーザに PreCacheDB を渡す。

2-2-5.PreCache Browser

PreCache Browser はユーザの要求を受けて、要求に沿った情報を PreCacheDB から検索し、ユーザに提示するブラウザである。また、ユーザからの要求に沿った情報を PreCacheDB が含んでいない場合、不足した情報を検索するためのキーをユーザから入力してもらい、そのキーを Deficiency Manager へ送り一時的に保持させる。PreCache Browser の例を図 5 に示す。

対象	img
カフェ メイリッシュ 種別: 喫茶店 最寄り駅: 秋葉原	
メイドダーツバー & カフェ LittlePSX 種別: 喫茶店 最寄り駅: 秋葉原	
鉄道居酒屋 LittleTGV 種別: 最寄り駅: 秋葉原	
秋葉原 ShikaDa屋 煮ると 種別: 居酒屋 最寄り駅: 秋葉原	
アキバ系 コスプレ居酒屋 小悪魔の宴 種別: 居酒屋	

図 5: PreCache Browser の例

2-2-6.Deficiency Manager

Deficiency Manager は PreCache Browser から不足した情報を検索するためのキーワードを取得・保持し、ネットワーク接続可能時に PreCache サーバへアクセス、不足した情報に対して再度 PreCache を作る要求を PreCache サーバへ送る。

3.プロトタイプ実装

本節では、PreCache の有用性を評価するためにプロトタイプを作成する。プロトタイプは名古屋市営地下鉄内の移動を移動想定範囲とし、PreCache する情報を「どの位置に何があるか」という位置依存情報として PreCacheDB を作成する。また、PreCache サーバは Java Applet で実装した。

PreCache Collector のプロトタイプは、名古屋市営地下鉄

の経路探索とその経路の途中にある駅のリスト化を実装した。また、名古屋市営地下鉄を使った移動に関する経路探索は出発駅からの駅数を幅とした、幅優先探索を用いた最短経路探索とした。名古屋市営地下鉄の路線図を図 6 に、経路探索に用いた幅優先探索の例を図 7 に示す。

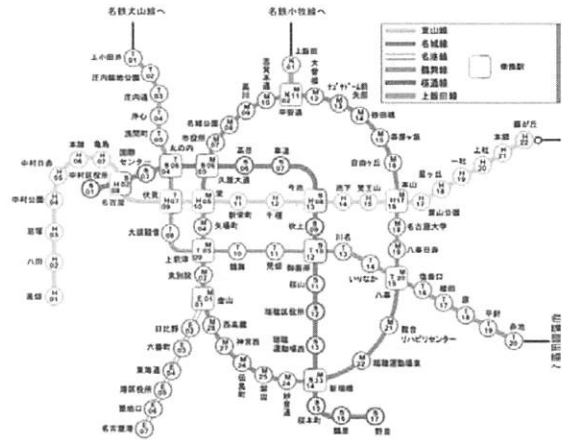


図 6: 名古屋市営地下鉄路線図

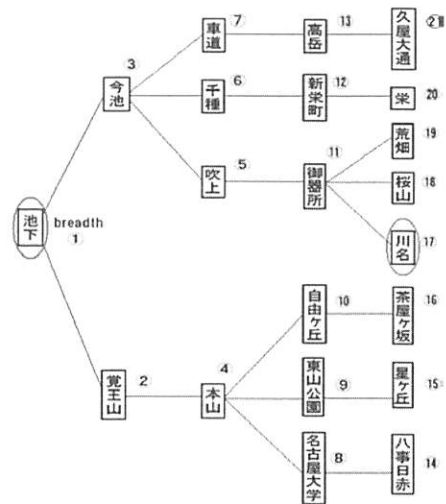


図 7: 池下駅から川名駅への幅優先探索の例

PreCache Crawler のプロトタイプは、ぐるなびから各駅周辺のレストラン情報を xml 形式で収集するモジュールを実装し、PreCache Converter のプロトタイプは、ぐるなびから収集してきた xml を解析し必要な情報を整形・抽出する機能を実装した。そして、PreCache Preserver のプロトタイプは、位置依存情報を格納するデータベーススキーマを実装し、PreCache Converter から渡されたぐるなびのレストラン情報を格納する機能を実装した。また、PreCache Browser は iPodTouch 及び iPhone アプリとして実装した。実装した PreCache Browser を図 8 に示す



図 8: PreCache Browser のプロトタイプ

4. 評価・考察

本節では、プロトタイプで作成した位置依存情報の PreCache を評価する。評価は Web ブラウザのキャッシュと PreCache について行う。

4-1. 関連研究・関連技術

Google Gears[3]

Google Gears は本来、オンラインでしか動かない Web アプリケーションをオフライン状態でも使えるようにするという発想で開発された。Google Gears の基本構成はローカルサーバ、データベース、ワーカプールの3つである。まず、Web アプリケーションをローカルサーバに保存しローカルホスト内でも動くようにする。次に、アプリケーションがデータの入出力を必要とする場合は、データをデータベースに格納する。そして、最後のワーカプールは、オンライン上のコンテンツをキャッシュし、コントロールする。これらの働きによって、オフラインでもユーザは Web アプリケーションを使えるようになる。ただし、Google Gears は Google Gears に対応した Web アプリケーションしかローカルホストで動かない点が問題として残る。

Supporting Cooperative Caching in Ad Hoc Networks[4]

この研究はアドホックネットワークを用いて、ユーザ達が持っているキャッシュを共有するというものである。アドホックネットワークを用いてキャッシュを共有できれば、ユーザが直接インターネットに接続できなくても周囲のユーザが持っているキャッシュから情報を検索できる可能性がある。また、同様の研究として、モバイルアドホックネットワークにおける効率的な情報共有[5]がある。

Web Cache

Web Cache は Web Proxy や Web ブラウザが、処理を高速

化するために一度ユーザが閲覧した Web の情報をキャッシュとして一定期間保存しておくものである。次に同じ Web を閲覧する要求が来た場合は、キャッシュを使うことによって処理を円滑に進める。無論、Web の情報は比較的早い頻度で腐るのでキャッシュは更新もされる。この Web Cache を使えばユーザが過去に見た Web ページの中から、情報を検索・利用できる可能性がある。

4.2 評価

PreCache の性能を評価するためには、オフライン時に PreCache を用いた情報検索とその他の手法での情報検索の性能を比較し評価する必要がある。よって評価は、検索された情報が妥当か、情報を検索するまでに煩わしい手間が発生しなかったか、検索できる情報の範囲はどうか、また、キャッシュする情報の量は妥当な量か、等で比較する。

評価すべき項目を以下に示す。

- ・適合性 (ユーザの要求にマッチする可能性)
- ・収集負荷 (情報を集めるまでの手間)
- ・ユーザビリティ (情報の検索しやすさ)
- ・情報量 (キャッシュする情報の量)
- ・スケーラビリティ (検索できる情報の範囲)

また、評価は各項目においてユーザが Web ブラウザを使って保存した Web キャッシュを使用して検索した場合と、PreCache を使用した場合とで比較をする。また、Web キャッシュを作るのはぐるなびの情報に限ることとした。結果は図 9 に記す。

項目 \ 種類	Web Cache	PreCache
適合性	高	高
収集負荷	大	小
ユーザビリティ	低	高
情報量	小	大
スケーラビリティ	大	小

図 9: Web キャッシュと評価

適合性について、Web キャッシュを用いて情報を検索する場合、キャッシュの情報にはユーザが事前に閲覧した Web の情報が保存されとおり、その後の行動で必要になりそうな情報が含まれている可能性が高い。これは一般的にユーザが移動をする前に、移動先の地図やその周辺で食事を取れる場所、電車の時間などを調べる事が多いと考えられるからだ。一方 PreCache の場合も、ユーザの移動に関して経路探索をして探索されたすべてのポイントの周辺情報を取得してくるので、ユーザの要求にマッチする可能性は高い。

収集負荷について、Web キャッシュを使う場合、ユーザが最低一度は情報を調べていなければならない。Web キャッシュの性質上、一度は Web を閲覧していないとキャッ

ッシュが作成されないからだ。この手間は大きい。しかし、PreCache の場合は、ユーザから入力された移動情報を元にほぼ自動で Web から情報を収集し、キャッシュを作成するため手間が少ないよって、Web キャッシュはユーザにとって負荷が大きく、PreCache はユーザにとって負荷が少ないと言える。

また Web キャッシュを使う場合、キャッシュから情報を検索するのは難しいためユーザビリティは低い。近年では過去の閲覧履歴などから検索ができるブラウザもあるが、どの情報をどの Web ページで見たかユーザ自身が覚えて限り、すぐに目的の Web キャッシュを発見できない。PreCache の場合、PreCache Browser で PreCache データベースから情報を検索できるため、ユーザビリティが高い。

情報量に関して、Web キャッシュはユーザが過去に閲覧した Web からキャッシュを作るため、全体の容量はさほど大きくない。一方、PreCache は位置情報をキーにして、Web から情報を Crawl してくるため情報量が大きくなる。また、Crawl して集めた情報はユーザが必要としない情報も多く含まれるため、情報量に対して利用価値のある情報が少なく効率が悪いとも言える。

最後にスケーラビリティに関して、Web キャッシュを使う場合、検索できる範囲はユーザが過去に閲覧したことがある Web ページに限られる。この為、スケーラビリティは低い。しかし、Web キャッシュの場合は過去に一度でも Web ページを閲覧していれば、その Web ページのキャッシュは作成されるので位置情報を制約とした PreCache を使うよりもスケーラビリティは高い。PreCache の場合は、位置情報を制約としているので、位置情報をキーとして収集した Web ページのキャッシュしかできない。よって、ユーザの位置情報をキーとした情報を検索する場合のスケーラビリティは高いが、そのほかの情報を検索する場合のスケーラビリティは低い。

4.3 考察

適合性・収集負荷・ユーザビリティについては、Web キャッシュよりも PreCache の性能が優れている。情報量についてはユーザが必要だと思い閲覧した Web ページのキャッシュとユーザが必要になりそうな情報を推測して Crawl し集めてくる PreCache では、PreCache の情報量が大きくなることは避けられない。しかし、ユーザが想定もしなかった情報を検索しようとした場合、Web キャッシュで情報を検索するのは難しく、PreCache の方が情報を検索できる可能性は高くなるので、情報量が大きくてもマイナス要因はさほどないと考えられる。スケーラビリティについては、Web キャッシュの方が Web 全体からユーザが閲覧した情報をキャッシュできるのに比べ、PreCache の方は PreCache Crawler が収集可能な範囲でしかキャッシュが取得できない。この点が Web キャッシュよりも PreCache の方が劣っている。これは、多くの PreCache Crawler を実装し、Web 上から多くの情報を収

集、ある程度スケーラビリティを向上させることができる。しかし、どれだけ PreCache のスケーラビリティを向上させても、Web キャッシュのスケーラビリティと比べると見劣りしてしまう。

以上のことから、PreCache は Web キャッシュに劣る部分もあるが、概ね PreCache の方が優れている事が分かる。

5.まとめと今後の課題

本稿では位置情報を制約条件として収集する PreCache を提案し、プロトタイプの実装と評価を行った結果、PreCache が有用であることを示した。しかし、プロトタイプ of PreCache は Web から情報を収集できる範囲が狭く、ユーザの位置に関する情報を抽出する範囲も名古屋市営地下鉄内に留まった。よって今後の課題は、Web から情報を収集できる範囲を広げるために PreCache Crawler を増やす事と、それに伴う PreCache Conveter の実装、またユーザの位置情報の取得に関しても、より幅広く経路探索を行えるようにする必要がある。

6.参考文献

- [1]総務省通信白書平成 20 年版、総務省情報通信統計データベース
- [2]ぐるなび、<http://www.gnavi.co.jp/>
- [3]Google Gears, Google, <http://gears.google.com/>
- [4] L. Yin and G. Cao, "Supporting Cooperative Caching in Ad Hoc Networks", *Proceedings of IEEE INFOCOM*, pp. 2537-2547, March 2004.,
- [5] 榎本 真, "モバイルアドホックネットワークにおける効率的な情報共有.", *ユニシス技報 Vol.27 No.4 通巻 95 号*, pp37-54, 2008.