論　文

Webパフォーマンス測定プラットフォームの提案と評価

中野　雄介[†,††a)]　上山　憲昭[†,††b)]　塩本　公平[††c)]　長谷川　剛[†††]
村田　正幸[†]　宮原　秀夫[†]

Proposal and Evaluation of Platform for Web Performance Measurement

Yuusuke NAKANO[†,††a)], Noriaki KAMIYAMA[†,††b)], Kohei SHIOMOTO[††c)], Go HASEGAWA[†††], Masayuki MURATA[†], and Hideo MIYAHARA[†]

あらまし　近年，Webパフォーマンスの重要性が注目を集めている．Webパフォーマンスとは，Webページ上のリンクがクリックされてから，次のWebページを構成するオブジェクトがダウンロードされ，表示が完了するまでの時間である．ユーザはWebパフォーマンスが低いWebページから離れる傾向にあり，Webパフォーマンスの低下はサービス提供者の収入の低下に直結する．このため，サービス提供者は自身が提供するWebページのパフォーマンスを測定し，パフォーマンス低下の原因を究明，改善する必要がある．しかし，このようなWebパフォーマンスの低下原因は，ネットワーク環境やクライアントの性能等，Webブラウザの動作環境によって変わると考えられる．本稿では，Webブラウザの動作環境を変化させ，且つ，Webパフォーマンスに加えてサーバ・ネットワーク・クライアントそれぞれのパフォーマンスについて測定可能とする，Webパフォーマンス測定プラットフォームを提案する．提案手法を評価した結果，提案手法は959個のWebページの88のホストでの測定において，測定にかかる稼働は30分程度でWebブラウザの動作環境の多様性によらず一定であり，PlanetLabを用いることで多様なWebブラウザの動作環境での測定ができることを確認した．これにより，測定結果から，Webページのパフォーマンス改善の方針を検討できるデータを収集することができた．

キーワード　Webパフォーマンス，測定環境，QoE，ボトルネック解析

1. はじめに

近年，Webパフォーマンスの重要性が注目を集めている．Webパフォーマンスとは，Webページ上のリンクがクリックされてから，次のWebページを構成するオブジェクトがダウンロードされ，表示が完了するまでの時間である．ユーザはWebパフォーマンスが低いWebページから離れる傾向にあり，Webパフォーマンスの低下はサービス提供者の収入の低下に直結する．例えば2000msの遅延はBingにおけるユーザあたりの収入を4.3%下げることがわかっている[1]．このため，サービス提供者は自身が提供するWebページのパフォーマンスを測定し，パフォーマンス低下の原因を究明，改善する必要がある．

Webパフォーマンスの低下の原因は3つに分類される．1つはWebサーバの性能に起因するものである．Webサーバによるリクエストの処理時間はWebパフォーマンスに直結する．2つ目はネットワークの性能に起因するものである．Webブラウザから Webサーバまでの RTT が大きい場合，Webパフォーマンスは大きく低下する傾向にある[1]．3つ目はWebブラウザやWebページの構成，PCの計算リソース等，クライアントに起因するものである．Webブラウザは，Webページを構成するスクリプトや画像などのオブジェクトをダウンロードし，それらを画面上に表示できる形に整形，ビットマップ化を行っており，このような処理は一般的にレンダリングと呼ばれる．ま

† 大阪大学大学院情報科学研究科　〒565–0871 大阪府吹田市山田2–1
Department of Information Science, Osaka University　1–5, Yamadaoka, Suita–shi, Osaka, 565-0871 Japan
†† 日本電信電話株式会社 NTT ネットワーク基盤技術研究所　〒180–8585 東京都武蔵野市緑町 3–9–11
NTT Network Technology Laboratories, NTT Corporation Midori–cho 3–9–11, Musashino–shi, Tokyo, 180–8585 Japan
††† 大阪大学サイバーメディアセンター　〒560–0043 大阪府豊中市待兼山町 1–32
Cybermedia Center, Osaka University　1–32, Machikaneyama, Toyonaka–shi, Osaka, 560-0043 Japan
a) E-mail: nakano.yuusuke@ist.osaka-u.ac.jp
b) E-mail: kamiyama.noriaki@ist.osaka-u.ac.jp
c) E-mail: shiomoto.kohei@lab.ntt.co.jp

た，Web ブラウザは，オブジェクトを並列にダウンロードすることで，Web パフォーマンスの向上を図っている．しかし，Web ページの構成によっては，効率的なレンダリングや並列ダウンロードができない場合があり，Web パフォーマンスに大きく影響する [2]．

ボトルネックとなる Web パフォーマンスの低下原因は Web ブラウザの動作環境によって変わると考えられる．例えば，クライアントマシンの処理性能が低いとクライアントがボトルネックとなる．一方，クライアントが RTT の長い国にある場合，ネットワーク性能がボトルネックとなる．このようなボトルネックを発見するためには，Web ページの構造 (HTML, JavaScript，CSS などの構造) を解析する静的解析ではなく，実際に Web ページを表示した際の各種測定値を解析する動的解析が必要となる．このため，様々な Web ブラウザの動作環境で，実際に Web ページを表示した際の Web パフォーマンスを測定するのみならず，Web パフォーマンス低下の原因を究明するために必要なデータも収集する必要がある．つまり，上記のような Web パフォーマンス低下の 3 つの原因のうち，ボトルネックとなる原因を特定するためには，サーバ・ネットワーク・クライアントそれぞれに関するパフォーマンスを測定する必要がある．

多様な環境で，多様なパフォーマンスを測定できるパフォーマンス測定 API として，近年 Navigation Timing [3] や Fathom [4] が提案されている．このような API が実装された Web ページは，ブラウザ内で様々な測定を行い，結果をサーバに送信する．多様なユーザ環境で，実際に閲覧された際のパフォーマンスを知ることができるため，ボトルネック発見には有用であると考えられる．しかし，専用の API を Web ページに実装する必要がある．一方，Web ブラウザには Web パフォーマンスを測定するためのツールが備わっており，任意の Web ページを表示した際の様々なパフォーマンスを測定することができる．しかし，このようなツールは人が直接操作することを前提としており，多様な環境での測定にはその多様性に比例した稼働を必要とする．

そこで本稿では，Web ブラウザの動作環境の多様性によらず一定の稼働量で，上記ような Web ブラウザの動作環境を変化させ，任意の Web ページの Web パフォーマンスに加えてサーバ・ネットワーク・クライアントそれぞれのパフォーマンスを測定する，Web パフォーマンス測定プラットフォームを提案する．提案プラットフォームにより，Web ページの提供者は，多様な Web ブラウザの動作環境で，自身の Web ページの Web パフォーマンスを測定可能となり，測定結果を分析することで，Web パフォーマンスの低下原因を特定し，改善することが可能となる．また，Web ページの提供者に加え，CDN 事業者や ISP 事業者にとっても有効であると考えられる．提案プラットフォームが Web パフォーマンスが低下している箇所についての情報を提供することで，CDN・ISP 事業者は，Web パフォーマンスの低い箇所に対して重点的に設備投資をすることができる．

2. Web パフォーマンス測定プラットフォームの要求条件

Web パフォーマンス測定プラットフォームは，Web パフォーマンス低下の原因を発見するために必要なデータを，多様な Web ブラウザ動作環境で収集する必要がある．また，Web ページの提供者が，自身の Web サーバでサービスを提供するとは限らないため，サーバ側でのデータ収集は困難な場合がある．もちろん，ネットワーク内でのデータ収集も困難である．このため，クライアント側で測定する必要がある．加えて，多様な Web ブラウザ動作環境のパターンで測定するためには，膨大な回数の測定が必要となり，測定のための稼働がかかるため，Web ブラウザの動作環境の多様性によらず一定の稼働量で測定できる必要がある．また，このような膨大な回数の測定は，途中で測定が失敗することも考えられ，常時監視と失敗時の復旧も含めた測定の自動化が求められる．

以下では，このような条件を満足するために Web パフォーマンス測定プラットフォームに求められる，測定データ，Web ブラウザ動作環境のバリエーションについて述べる．

2.1 測定データ

まず，Web パフォーマンス測定プラットフォームは，Web パフォーマンスを測定する必要がある．Web パフォーマンスとしては，onload 時間を用いるのが一般的である．onload とは，Web ブラウザが画面に表示するために必要なオブジェクトを全てダウンロードしたタイミングで発行するイベントであり，Web ページを構成するオブジェクトのダウンロード開始から，onload までの時間を Web パフォーマンスと考える．なお，一般的な Web ブラウザは，オブジェクトのダウンロードと画面の描画は並列に実行されるため，

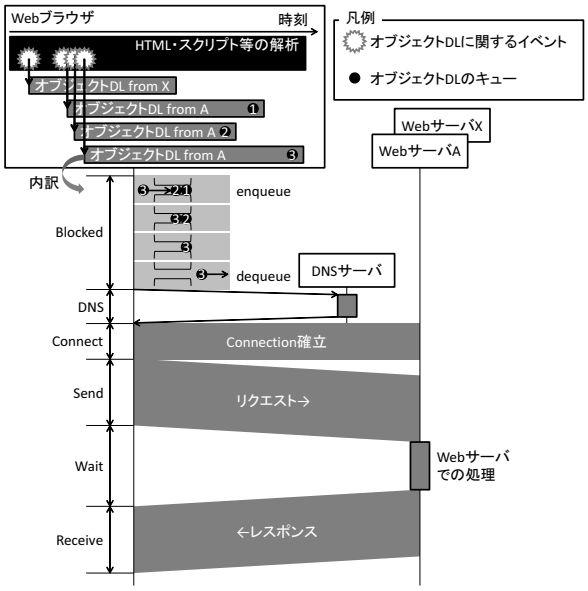

図 1　オブジェクトのダウンロードにかかる時間の内訳
Fig. 1　Breakdown of download time

onload イベントが発行される時刻と画面の描画完了が必ずしも一致しているわけではない．

加えて，Web パフォーマンス測定プラットフォームは，Web パフォーマンス低下の原因箇所を特定するためのデータも収集する必要がある．Web ページを構成するオブジェクトは HTTP でダウンロードされ，各オブジェクトのダウンロードにかかる時間は，ダウンロード中の処理内容によって幾つかに分離できる．このようなダウンロードの内訳を収集できれば，Web パフォーマンス低下の原因究明に有効であると考えられる．Firefox や Chrome といった近年の Web ブラウザはこのような内訳ごとに時間を測定でき，測定結果は HAR(HTTP Archive) [5] と呼ばれる JSON ファイルとして出力できる．以下で HAR で収集できる内訳について説明する．

図 1 は HAR 形式で記録される各内訳を図示したものである．Web ブラウザは Web ページを構成する HTML や各種スクリプトを解析すると同時に，解析結果にしたがって，Web ページの表示に必要なスクリプトや画像などのオブジェクトをダウンロードするためのイベントを発行する．このイベントを契機としてそれぞれのオブジェクトのダウンロードが開始される．図 1 の例では，サーバ X に対するオブジェクトのダウンロードのイベントが発生し，その後，立て続けにサーバ A に対する 3 つのダウンロードのイベントが発生する．このように，オブジェクト間には依存関係があり，その依存関係に従った順番でオブジェクトはダウンロードされる．また，単一の Web サーバから複数のダウンロードを行う際は，一定の並列数に制限される．この例では，並列数は 1 とされており (Chrome では 6 に設定されている)，1 つのオブジェクトがダウンロードされている間，後のダウンロードはエンキューされ，デキューされるまで待つ．これは Blocked と呼ばれる．なお，Blocked の原因には様々なものがあり，文献 [6] で紹介されている．その後デキューされ，ダウンロードが開始される．まず，Web サーバの IP アドレスを取得するため，DNS サーバを用いて名前解決するための時間がかかる．DNS キャッシュされている場合は，この時間は 0 となる．Web サーバの IP アドレスが判明すると，Web ブラウザはサーバとコネクションを確立し，HTTP リクエストを送信し，HTTP レスポンス受信まで待ち，HTTP レスポンスを受信する．これらはそれぞれ Connect，Send，Wait，Receive と呼ばれ，それぞれに時間がかかる．なお，以上のような内訳の他に，onload 時間についても HAR に記録される．

このようにして分離された各内訳は，関連する箇所 (サーバ・ネットワーク・クライアント) がある程度決まっているため，オブジェクトのダウンロードの際に最も時間のかかる内訳がわかれば，Web パフォーマンス低下の原因を含む箇所を特定することができる．つまり，DNS，Connet，Send，Receive はネットワークに関する時間である．一方，Wait はサーバでの処理時間と，ネットワークでの RTT 両方が含まれる．このため，Wait をサーバとネットワークそれぞれにかかる時間に分離するため，RTT についても測定する必要がある．また，Blocked はブラウザでの処理における待ち時間であるため，クライアントに関する時間であるが，他のオブジェクトのダウンロードの待ち時間も含まれるため，ネットワークに関する時間でもある．このため，クライアントに関する値として，クライアントマシンのベンチマークスコアも収集する．

以上から，Web パフォーマンス測定プラットフォームは測定データとして onload，Blocked，DNS，Connet，Send，Wait，Receive の時間 (HAR 形式) ，RTT，クライアントマシンのベンチマークスコア を収集する必要がある．

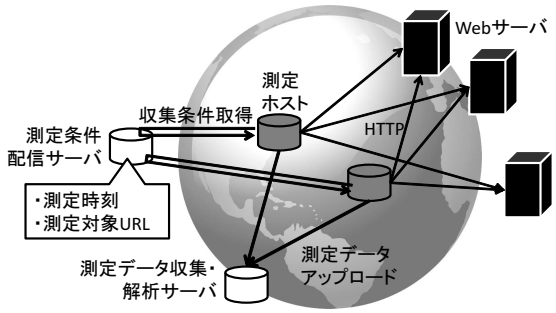

図 2　Web パフォーマンス測定プラットフォームの概要
Fig. 2　Outline of platform for web performance measurement

2.2　Web ブラウザの動作環境

Web パフォーマンスの低下原因は Web ブラウザの動作環境によって変化する．また，Web ページは多様な環境で動作する Web ブラウザによって表示される．このため，Web ページの提供者は，多様な Web ブラウザの動作環境において，自身の Web ページのパフォーマンスを測定する必要がある．Web ブラウザの動作環境を決定する要素を以下に列挙する．

ネットワーク内でのクライアントの位置　クライアントの位置によって，Web パフォーマンスは変化する．これは，回線の性能 (RTT とスループット) と，ダウンロードするオブジェクトとの距離が変化するためである．例えば，クライアントが南米にある場合，北米の Web サーバが提供する Web ページに対する Web パフォーマンスは，クライアントが北米にある場合と比較して悪くなる．

時刻　Web ページを表示する時刻や曜日によっては，輻輳等の発生により Web パフォーマンスが低下することがある．

クライアントの処理性能　クライアントの計算リソースが少ない場合は，スワップの発生等により Web ページのレンダリングに時間がかかり，Web パフォーマンスが低下すると考えられる．

以上の要素の組み合わせにより，Web パフォーマンス測定プラットフォームは多様な Web ブラウザの動作環境を実現する必要がある．

3.　Web パフォーマンス測定プラットフォームの提案

以上のような要求条件を満たす Web パフォーマンス測定プラットフォームを提案する．図 2 に提案プラットフォームの概要を示す．提案プラットフォームは世界中に配置された測定ホスト，測定対象の URL や測定開始時刻等の測定条件を保持・配信する測定条件配信サーバ，測定データ収集・解析サーバから構成される．

今回，測定ホストには，インターネットに接続された世界中のホストを自由に利用できる実験環境である，PlanetLab [7] を用いた．各地のホストに Web ブラウザと測定プログラムをインストールし，測定用ホストとした．PlanetLab のホストは，様々な大学や企業が自身の保持するホストマシンを，バーチャルマシンとして提供することで成り立っている．提供されるホストマシンの処理性能は様々であるため，予め各ホストでベンチマークソフトを走らせ，ホストごとの処理性能を測定，収集しておく．なお，ベンチマークソフトとしては，UnixBench を用いた．

PlanetLab は一般的な Linux がインストールされたバーチャルマシンを提供している．このため，PlanetLab 以外の一般的な VPS や独自のホストを利用してもよい．この場合，世界各地に様々な処理性能のホストを自身で配置することになる．

測定条件配信サーバは Web サーバであり，測定対象の URL と測定開始時刻等の測定条件とを保持・配信する．測定ホストにインストールされた測定プログラムは，測定条件配信サーバから，測定対象 URL と測定条件とを取得し，それらに従って Web パフォーマンスを測定する．

測定データ収集・解析サーバは FTP サーバであり，各測定ホストで収集された HAR と RTT のデータは FTP を介して測定データ収集・解析サーバにアップロードされる．

3.1　測定ホスト

図 3 に測定ホストの概要を示す．測定ホストには，予め測定プログラムと Firefox がインストールしておく．測定プログラムは Cron などによって起動される．その後，測定プログラムは，測定条件配信サーバから HTTP を介して測定条件を取得する．測定条件が公開されていない場合，測定プログラムは測定条件が公開されるまで待つ．測定条件には測定開始時刻が記載されており，測定プログラムはその時刻まで待機する．なお，測定ホストのある国の現地時刻での測定開始時刻まで待機する必要があるが，PlanetLab のホストの時刻は全て UTC に設定されている．このため，GeoNames [8] が提供するデータを用い，測定ホスト

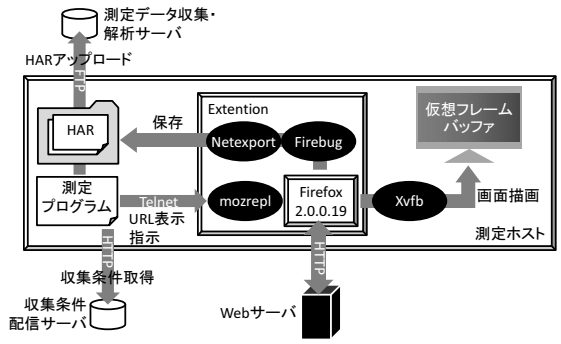

図 3　測定ホストの概要
Fig. 3　Outline of measurement host

の位置情報 (測定条件配信サーバによって配信される) から現地時刻を特定し，その時刻から待機する時間を算出する．

待機時間が満了すると，測定プログラムは Web ブラウザを起動・操作する．今回は Web ブラウザとして Firefox を用いた．Firefox のバージョンは 2.0.0.19 であり，キャッシュをローカルに保存しない設定としている．測定プログラムは，Firefox を Telnet 経由で遠隔操作する Extention である mozrepl を用い，Firefox に測定条件に含まれる測定対象の URL を表示するよう指示する．なお，Firefox 以外の Web ブラウザを用いる場合，遠隔操作のためのプログラムや HAR の生成方法を，Web ブラウザにあわせて選定する必要がある．Firefox は該当する URL の Web サーバに対して HTTP リクエストを送信し，その Web ページを表示する．PlanetLab ノードには画面が存在しないため，Xvfb を介して仮想フレームバッファにレンダリング結果を描画することで，実際の Web ブラウザの動作を実現している．PlanetLab 以外の仮想サーバ等を利用する場合も画面がないと考えられるため，同様の手法を用いることができる．HAR の生成には，Web ページのデバッグ等の機能を提供する Extention である Firebug と，Firebug 上で動作する HAR 生成用の Extention である Netexport を用いる．

測定プログラムは HAR が生成されると，生成された HAR を解析し，Web ページを構成するオブジェクトを提供する Web サーバのホスト名を抽出する．その後，各ホストに対して 1 度ずつ Ping を送信し，RTT を測定し，測定結果をファイルに保存する．このようにして，測定プログラムは HAR と RTT の測定結果を生成する．

最後に，測定プログラムは生成された HAR ファイルと RTT の測定結果のファイルとを測定データ収集・解析サーバに FTP でアップロードする．同様にして，測定プログラムは，収集対象の全ての URL に対する HAR と RTT とを生成し，アップロードする．

なお，Firefox のプロセス停止などにより，測定不能となることがある．測定プログラムは HAR ファイルの生成状況を監視し，HAR が生成できていない場合，Firefox の動作が不完全であると判断し，復旧を図る．復旧には予め作成しておいた，正常に動作する Firefox の設定ファイル等を用いる．測定プログラムは Firefox の動作不全を発見すると，Firefox のプロセスを終了し，正常に動作する Firefox の設定ファイルを上書きし，Firefox を再起動する．これにより，Firefox を再び正しく動作させることができる．また，FTP サーバとのコネクションが切断された場合も，再接続を行う．

3.2　測定条件配信サーバ

測定条件配信サーバは，測定プログラムに対して測定条件を配信する Web サーバである．測定条件には測定対象の URL と測定開始時刻等が含まれる．測定条件の例を図 4 に示す．測定条件は JSON 形式で記述される．例では 2 行目に測定ホスト内で測定データが保存されるディレクトリのパス，3 行目に Firefox を起動するためのパス，4 行目に測定ホストの緯度経度を配信する URL，5 行目に Firefox が正しく動作するコンフィグファイル等のパスが記述される．また，6〜10 行目には測定データ収集・解析サーバに測定データをアップロードするための FTP サーバについての情報が記述される．加えて，11 行目以降で複数の測定スケジュールを設定できる．測定スケジュールは測定開始時刻，測定時のタイムアウト時間，測定対象の URL リストを配信する URL で構成される．これらにより，測定プログラムは指定された現地時刻で，対象 URL の Web パフォーマンスを測定できる．また，Firefox の起動・再起動，測定データのアップロードが可能となる．

3.3　Web パフォーマンス測定プラットフォームと要求条件との対応

以上のような Web パフォーマンス測定プラットフォームは，先に挙げた要求条件を満足するものとなる．まず，測定データについては，Firefox を用いた HAR の収集と，各 Web サーバに対する RTT の測定，各測定ホストのベンチマークスコアの測定により，要

```
{
"path_to_har":"/home/test/logs/",
"path_to_firefox":"/home/test/firefox",
"location_url":"http://osakafnet.com/location.txt",
"path_to_firefox_targz":"/home/test/env.tar.gz",
"ftp":{
"host_name":"osakafnet.com",
"port":21,
"dir_name":"data/"
},
"timing":[
{
"start":"0:0:0",
"timeout":30,
"target_list_url":"http://www.osakafnet.com/url.txt"
},
   :
   :
}
```

図 4　測定条件の例
Fig. 4　Example of measurement configuration file

求条件を満足している．次に，Web ブラウザの動作環境については，PlanetLab を用いることで，多様な位置，多様な処理性能のホストを用いることができ，動作環境の多様性を確保することができる．また，測定時刻に関しても，測定条件配信サーバにスケジュールを設定することで，各地点の現地時刻での多様な時刻に測定を開始できる．このような多様な Web ブラウザの動作環境において，動作環境の多様性によらず一定の稼働量で測定可能である．加えて，Firefox や FTP の不調時には自動復旧されるため，稼働をかけることなく，測定プラットフォームの可用性を高めている．

4. 評　　価

提案プラットフォームの有効性を評価するため，実際に公開されている Web ページの Web パフォーマンスを測定することで，収集にかかる稼働，Web ブラウザ動作環境の多様性，可用性を確認した．加えて，収集された測定結果を解析することで，Web パフォーマンスの改善方針を検討できることを確認した．

4.1　測定にかかる稼働の評価

測定を開始するには以下の作業が必要である (カッコ内は作業時間の目安)．

（1） 測定プログラムを各測定ホストにアップロード (5 分程度)

（2） 測定条件を作成 (条件の内容に依存)

（3） 測定ホストの cron を設定 (10 分程度)

(1)，(3) については，pssh を用いることで複数の測定ホストに対して同時にコマンド実行し，作業にかかる時間を短縮している．また，測定条件については，図 4 のファイルと，URL を羅列したファイルを作成し，収集条件配信サーバにアップロードすればよい．なお，1 回の測定のためのこれらの設定にかかる時間は，測定条件にもよるが，Web ブラウザの動作環境の多様性によらず，概ね 30 分程度である．

4.2　可用性の評価

可用性を評価するため，長時間の測定の完了ホスト数を確認した．長時間の測定のために，多くの URL を測定対象とする必要がある．また，より実際の測定に近づけるため，実際にサービスを提供しており，多くの人が利用する様々な Web ページを対象とする必要がある．このため，Alexa [9] のランキング上位の Web ページの URL を測定対象とした．なお，様々な Web ページを対象とするために，Web ページのカテゴリ (ニュースやショッピングなど) ランキング上位から同数ずつ抽出し，959 個の URL リストを作成した．

なお，1 回の測定のタイムアウト時間は 60 秒とし，タイムアウト時間を満了するなど，測定に失敗した場合は，1 回リトライする条件とした．これは，これまでの onload 時間の測定から，多くの Web ページの onload 時間は 20 秒以内であり，60 秒程度のタイムアウト時間でページの表示の失敗を検出できると考えたためである．また，リトライはタイミングによる問題で失敗した測定をやり直すためのものであり，1 回以上のリトライで測定できない URL は，根本的な問題により測定できないと判断するため，リトライ回数は 1 回とした．

以上の測定を 88 の測定ホストにおいて実施した．この結果，測定が完了したホスト数は 78 であった．測定が完了できなかった 3 ホストについては，PlanetLab によってリソースの専有と判定され，プロセスが停止されていた．また，残りの 6 ホストについては，Firefox の再起動に失敗しており，1 ホストについては動作が不安定なホストであったため，何らかの理由で測定プログラムが停止したと考えられる．

4.3　測定成功数の評価

959 個の URL のうち，測定に成功した個数について，平均値： 675，中央値： 698，最小値： 286，最大値： 801 という結果となり，概ね 7 割程度の Web ページで測定に成功していることがわかる．失敗する Web ページについては，HAR の収集や RTT の測定

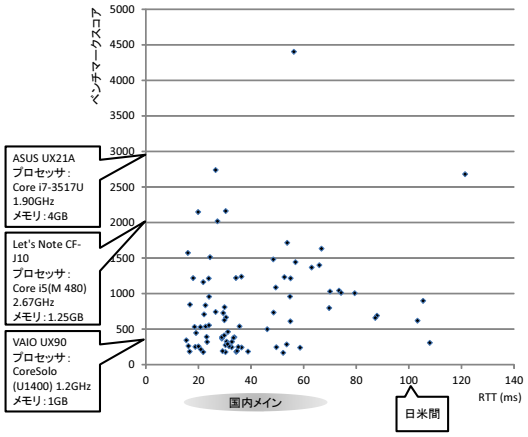

図 5　測定ホストの RTT とベンチマークスコアの分布
Fig. 5　Scatter diagram of RTT and benchmark score of measurement hosts

がタイムアウト時間満了までに完了していないと考えられる．

4.4　Web ブラウザの動作環境の多様性

提案プラットフォームは，ネットワーク内でのクライアントの位置，測定開始時刻，クライアントの処理性能といった条件の組み合わせにより，多様な Web ブラウザの動作環境を提供する．図 5 に測定ホストごとの RTT とベンチマークスコアの分布を示す．RTT は先の 959 個の URL の Web ページを表示する際にオブジェクトをダウンロードする全ての Web サーバに対する RTT の平均である．

図 5 の横軸は RTT を表し，日本からの RTT と大まかな距離との関係を例示している [10]．主に国内のサーバからオブジェクトをダウンロードするホストから，遠くのサーバからも多くのオブジェクトをダウンロードするホストまで，多様な RTT のホストで測定できていることがわかる．また，縦軸は測定ホストのベンチマークスコアと具体的なパソコンのスペックを例示している [11]．一般的なユーザが利用するノートパソコンで，数年前に発売されたものから最近発売されたものまで，多様なスペックに対応するホストで測定できていることがわかる．以上から，提案プラットフォームは多様な Web ブラウザの動作環境を提供できていることがわかる．

なお，測定時刻については測定条件に記述される測定開始時刻に多様な時刻を記述することができる．

4.5　測定データの解析結果の例

以上の測定において収集されたデータの解析結果の例として，サーバ，ネットワーク，クライアントそれぞれについての測定値と Web パフォーマンスとの相関係数に基づく Web パフォーマンス低下の原因となるボトルネックの特定と，その結果に基づく Web パフォーマンス改善方針の検討を行った．

Wang らは，Web ページのレンダリングでは，複数のオブジェクトのダウンロード，HTML のパース，JavaScript や CSS の評価を並列で行っているため，これらの処理のうちクリティカルパス上にある処理がボトルネックであり，このような処理を削減することが Web パフォーマンスの向上に繋がると述べている [6]．つまり，クリティカルパス上にある処理の増減は，onload 時間の増減に繋がると考えられるため，onload 時間と相関の高い測定値と関連する箇所がボトルネックであると考えられる．

今回の測定データを用い，上記の傾向があることを確認した．表 1 に Web ブラウザ内での処理量の大小と，onload 時間とベンチマークスコア，RTT，サーバでの処理時間との間の相関係数とを示す．ここでの処理量とは HTML，CSS，JavaScript の合計サイズ

表 1　ブラウザ内での処理量の大小と，onload 時間とベンチマークスコア (BM)，RTT，サーバでの処理時間との間の相関係数との関係
Table 1　Relationship between amount of processing in web browser and correlation coefficient between onload time and benchmark(BM) score, RTT and processing time of web server

	onload 時間との相関係数		
URL	BM	RTT	Server
処理量が多いページ			
www.yr.no	-0.60	-0.20	0.21
www.4shared.com	-0.27	-0.07	0.70
www.webmd.com	-0.63	-0.10	0.30
games.yahoo.com	-0.51	-0.26	0.78
www.cvs.com	-0.50	-0.18	0.31
www.victoriassecret.com	-0.50	-0.03	0.62
www.about.com	-0.53	-0.08	0.47
www.jw.org	-0.33	-0.05	0.49
www.elsevier.com	-0.52	-0.05	0.34
www.xcams.com	-0.54	0.05	0.63
処理量が少ないページ			
www.hentai-foundry.com	-0.14	0.17	0.88
multiply.com	0.17	0.24	0.72
www.asstr.org	-0.05	0.33	0.78
manoramaonline.com	0.19	-0.03	0.95
southern-charms.com	-0.15	0.01	0.99
www.rediff.com	-0.36	-0.04	0.47
thumblogger.com	-0.59	0.31	0.33
copyscape.com	0.01	0.17	0.71
payserve.com	-0.15	0.13	0.85
thepiratebay.se	-0.01	0.30	0.16

のことである．処理量の多いページとは，測定対象のURL のうち，上位 10 の処理量のページとした．同様に，処理量の少ないページとは，測定対象の URL のうち，下位 10 の処理量のページとした．処理量の多いページはレンダリングのためのオブジェクトのパースや評価の処理が多くなり，このようなレンダリング処理がクリティカルパスの多くを占めるため，レンダリング処理がボトルネックとなる．このような処理にかかる時間はクライアントの性能によって増減するため，処理量の多い (レンダリング処理がボトルネックの) ページにおいては，ベンチマークスコア (BM) と onload 時間との相関係数が大きくなると考えられる．

表 1 を参照すると，処理量の多いページでは，onload 時間とベンチマークスコアとの相関係数の絶対値が，処理量の少ないページと比較して大きいことが分かる．また，処理量の少ないページでは，RTT やサーバでの処理時間といったオブジェクトのダウンロード時間と関連する値と onload 時間との相関係数の絶対値が大きくなることも分かる．このことから，onload 時間との相関係数が大きい測定値と関連する箇所がボトルネックであると考えられる．

以上から，Web パフォーマンス低下の原因となるボトルネックの特定のため，まず，各 Web ページについて，測定結果から，測定地点毎に onload 時間，サーバでの処理時間，RTT，ベンチマークスコアのデータを抽出した．その後，Web ページごとに全測定地点の onload 時間とサーバでの処理時間との間の相関係数を算出した．同様にして，onload 時間と RTT，onload 時間とベンチマークスコアとの間の相関係数も算出した．最後に，相関係数が 0.4 以上で正の相関有り，-0.4 以下で負の相関有り，それ以外を無相関として，各相関の Web ページの数を集計した．表 2 に集計結果を示す．表の各値はそれぞれの相関を持つ Web ページの数を表し，例えば onload 時間と RTT との間に負の相関のある Web ページは 7 個であることがわかる．

表 2　onload 時間と RTT，サーバでの処理時間，ベンチマークスコアとの間における各相関の Web ページの個数 (相関係数の絶対値が 0.4 以上で相関有りと判定)

Table 2　Number of web pages which have each kind of correlation between onload time and RTT, onload time and server time, and onload time and benchmark score (consider absolute value of correlation coefficient which is greater than 0.4 as corelative)

測定値＼相関有無	正の相関	無相関	負の相関
サーバ処理時間	506	226	1
RTT	55	671	7
ベンチマークスコア	9	472	252

このような onload 時間とサーバでの処理時間，RTT，ベンチマークスコアとの相関係数を算出することで，サーバでの処理時間，RTT，クライアントの性能の中で，何が Web パフォーマンス (onload 時間) に影響を与えるかを特定することができ，ある程度の Web パフォーマンスの改善方針の検討が可能となる．例えば，onload 時間とベンチマークスコアとに負の相関があれば，クライアントの計算リソースによって Web パフォーマンスが左右されていることがわかる．つまり，クライアントでの Web ページのレンダリング処理がボトルネックとなっており，Web ページの構造を改善することとなる．また，onload 時間と RTT とに正の相関があればオブジェクトのリクエストや転送がボトルネックとなっていると考えられ，CDN の利用などを検討することとなる．一方，onload 時間とサーバでの処理時間とに正の相関があれば，サーバでの処理時間がボトルネックになっていると考えられ，サーバの処理性能の増強などを検討することとなる．

以上のように，多様な Web ブラウザの動作環境で Web パフォーマンスを測定することで，Web ページのボトルネックを発見することができる．また，上記で使用した測定値以外の値も利用することで，更に詳細な解析も可能である．

5. 関 連 研 究

Web パフォーマンスの測定と，ボトルネックの発見については様々な研究が行われてきた．

Meenan は Web パフォーマンスのアクティブテスト手法を体系的に紹介している [12]．また，実践的な Web パフォーマンス向上手法がまとめられた参考書として，Grigorik による High Performance Browser Networking [1] や Souders による High Performance Web Sites [2] が有名である．

Web パフォーマンスを測定するツールも登場しており，YSlow [13] や Chrome DevTools [14] は広く利用されているほか，自身が提供する Web サイトが実際にユーザのブラウザで表示される際のパフォーマンスを測定するための API として Navigation Timing [3] や Dhawan らは Fathom [4] などがあり，Web パフォーマンスの測定環境は充実してきている．Fathom で Web ページを実装することで，その Web ページを表示し

た際に，Web ブラウザは各種測定結果をサーバに送信する．これにより，Web ページの提供者は，実際にページが表示される際のパフォーマンスを知ることができる．加えて，Web パフォーマンスの静的解析ツールも開発されており，Google が提供する PageSpeed Insights は広く利用されている [15]．Wang らは Web ページを構成するオブジェクト間の依存関係の種類を分類・定義し，それに基づき Web ページを構成するオブジェクト間の依存グラフを生成することで Web パフォーマンスのボトルネックを発見するプロファイラである WProf を開発している [6]．このようなツールは，個々の測定のために人が操作することを前提に作られているため，Web ブラウザの動作環境の多様性や，対象の Web ページの多様性に比例した稼働が発生する．一方，提案プラットフォームは，一定量の測定前の準備が必要であるが，その後は自動で測定が行われ，測定のための稼働量は Web ブラウザの動作環境や対象の Web ページの多様性によらず一定である．

様々な場所から Web パフォーマンスを測定する環境についても研究されている．Sundaresan らは各家庭のブロードバンドルータに測定プログラムをインストールすることで，有名な 9 個の Web ページのパフォーマンスを測定し，ボトルネックを特定した [4]．

以上のように，様々な Web パフォーマンス測定，ボトルネック発見手法について研究されているが，様々な Web ブラウザ動作環境で，特定の API が実装されていない任意の Web ページのパフォーマンスを，Web ブラウザの動作環境の多様性によらず一定の稼働量で動的解析するための手法は提案されていない．

6. おわりに

本稿では，Web ブラウザの動作環境を変化させ，且つ，Web パフォーマンスに加えてサーバ・ネットワーク・クライアントそれぞれのパフォーマンスについても測定可能とする，Web パフォーマンス測定プラットフォームを提案した．提案手法を測定にかかる稼働，可用性，測定成功数，測定時間，Web ブラウザ動作環境の多様性，測定データの解析結果の観点から評価した．この結果，提案手法は 1 回の測定にかかる稼働は 30 分程度で Web ブラウザの動作環境の多様性によらず一定であった．また，959 個の Web ページを 88 のホストで測定した結果，78 のホストで測定を最後まで継続できることを確認した．一方，PlanetLab を用いることで，多様な Web ブラウザ動作環境での測定を実現した．これにより，測定結果から，Web ページのパフォーマンス改善の方針を検討できるデータを収集することができた．

今後は，モバイル環境における Web パフォーマンス測定のため，提案プラットフォームを拡張する．

文　献

[1] I. Grigorik, High Performance Browser Networking What every web developer should know about networking and web performance, O'Reilly Media, 2013.

[2] S. Souders, High Performance Web Sites Essential Knowledge for Front-End Engineers, O'Reilly Media, 2007.

[3] "Navigation timing 2". http://www.w3.org/TR/navigation-timing-2/

[4] M. Dhawan, J. Samuel, R. Teixeira, C. Kreibich, M. Allman, N. Weaver, and V. Paxson, "Fathom: A browser-based network measurement platform," Proceedings of the 2012 ACM Conference on Internet Measurement Conference, pp.73–86, IMC '12, ACM, New York, NY, USA, 2012.

[5] "Http archive (har) format". https://dvcs.w3.org/hg/webperf/raw-file/tip/specs/HAR/Overview.html

[6] X.S. Wang, A. Balasubramanian, A. Krishnamurthy, and D. Wetherall, "Demystifying page load performance with wprof," Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13), pp.473–485, USENIX, Lombard, IL, 2013.

[7] "Planetlab". https://www.planet-lab.org/

[8] "Geonames". http://www.geonames.org/

[9] "Alexa". http://www.alexa.com/

[10] 田中裕之，高橋紀之，川村龍太郎，"Iot 次代を拓くエッジコンピューティングの研究開発," NTT 技術ジャーナル，vol.27，no.8，pp.59–63，2015.

[11] "Pc/linux benchmark". http://www.naoya.net/wiki/pc/linux-benchmark

[12] P. Meenan, "How fast is your website?," Commun. ACM, vol.56, no.4, pp.49–55, April 2013.

[13] "Yslow". http://yslow.org

[14] "Chrome devtool". https://developer.chrome.com/devtools

[15] "Pagespeed insights". https://developers.google.com/speed/pagespeed/insights/

（平成 xx 年 xx 月 xx 日受付）

中野　雄介　（正員）

平成 17 年和歌山大学大学院システム工学研究科修了．同年日本電信電話株式会社入社．以後，NTT ネットワークサービスシステム研究所，NTT ネットワーク基盤技術研究所勤務．Web スクレイピング，ユビキタスコンピューティング，サービスデリバリープラットフォーム，分散データベース，Web パフォーマンス等の分野の研究に従事．平成 25 年大阪大学大学院情報科学研究科招へい教員，平成 27 年同招へい准教授．博士 (情報科学) (平成 23 年 3 月，大阪大学)．

上山　憲昭　（正員）

平成 8 年 3 月，大阪大学大学院工学研究科通信工学専攻博士後期課程修了．工学博士．平成 8 年 4 月より，南カリフォルニア大学 訪問研究員．平成 9 年 4 月，日本電信電話株式会社入社．以後，NTT マルチメディアネットワーク研究所，NTT サービスインテグレーション基盤研究所，NTT ネットワーク基盤技術研究所勤務．平成 25 年 4 月，大阪大学大学院情報科学研究科 招へい准教授，平成 27 年 4 月，同招へい教授．ネットワーク設計，ネットワーク制御，トラヒック制御，トラヒック測定，コンテンツ配信ネットワーク，ネットワーク経済性，等の分野の研究に従事．IFIP/IEEE IM 2013 Best Paper Award を受賞．

塩本　公平　（正員：フェロー）

日本電信電話（株）ネットワーク基盤技術研究所　通信トラヒック品質プロジェクト　プロジェクトマネージャ．昭和６２年、阪大・基礎工・情報工卒、平成元年、阪大・基礎工・情報工修了．平成元年４月、日本電信電話（株）入社．以来、平成８年７月まで、ATM ノードシステムの開発および ATM トラヒック制御技術の研究開発に従事．平成８年８月より平成９年９月まで、米国ワシントン大学（ミズーリ州セントルイス）にて客員研究員として、高速ネットワークアーキテクチャの研究開発に従事．平成９年１０月より平成１３年６月まで、日本電信電話（株）ネットワークサービスシステム研究所にて、IP/MPLS ラベルスイッチルーターの研究開発に従事．平成１３年７月より平成１６年３月まで、日本電信電話（株）未来ねっと研究所にて、光ルーターの研究開発に従事．平成１６年４月より現在に至るまで、日本電信電話（株）ネットワークサービスシステム研究所にて、IP オプティカルネットワーキングに関する研究開発に従事．平成 23 年 7 月より日本電信電話（株）ネットワーク基盤技術研究所にてトラヒックエンジニアリングの研究開発に従事，平成 24 年 7 月より日本電信電話（株）ネットワーク基盤技術研究所にて通信トラヒック品質の研究開発に従事．平成７年に、電子情報通信学会学術奨励賞を受賞．平成７年、１３年に、電子情報通信学会交換システム研究賞を受賞．著書に「やさしい MPLS」（電子通信協会）および「GMPLS Technologies: Broadband Backbone Networks and Systems (Optical Engineering)」（Marcel Dekker Inc）．電子情報通信学会フェロー、IEEE シニア会員、ACM 会員．阪大・博士（工学）．

長谷川　剛　（正員）

平成 7 年大阪大学基礎工部情報工学科退学．平成 9 年同大学大学院博士前期課程修了．同年同博士後期課程退学．同年大阪大学経済学部助手．平成 10 年同大大学院経済学研究科助手．平成 12 年同大サイバーメディアセンター助手．平成 14 年同助教授．現在，大阪大学サイバーメディアセンター准教授．博士（工学）．主としてトランスポート層プロトコル，オーバレイネットワーク，無線ネットワーク，ネットワーク計測などの研究に従事．情報処理学会，IEEE 会員．

村田　正幸　（正員：フェロー）

昭和 57 年大阪大学基礎工学部情報工学科卒業．昭和 59 年同大学大学院博士前期課程修了．同年日本アイ・ビー・エム（株）入社．同社東京基礎研究所を経て，昭和 62 年大阪大学大型計算機センター助手．平成 1 年同大学基礎工学部助手．平成 3 年同講師．平成 4 年同助教授．平成 11 年同教授．平成 12 年大阪大学サイバーメディアセンター教授．現在，大阪大学大学院情報科学研究科教授．工学博士．システムのモデル化と性能評価，情報ネットワークアーキテクチャなどの研究に従事．IEEE，ACM 会員．

宮原　秀夫　（正員：フェロー）

昭和 42 年大阪大学工学部通信工学科卒．昭和 48 年大阪大学工学博士．昭和 58 年 IBM トーマス・ワトソン研究所客員研究員．平成元年大阪大学大学院基礎工学部教授．平成 7 年大阪大学計算機センター　センター長．平成 9 年大阪大学基礎工学研究科教授．平成 10 年大阪大学基礎工学研究科科長・基礎工学部長．平成 14 年大阪大学情報科学研究科教授・研究科長．平成 15 年大阪大学総長．平成 19 年情報通信研究機構理事長．平成 21 年（一社）臨床医工情報学コンソーシアム関西会長．現在大阪大学名誉教授、将来ネットワーク共同研究講座　特任教授．

Abstract Web performance is getting important in recent years. Web performance means the time from clicking a link on a web page to finishing showing the web page of the link and low web performance web pages are tend to lose customers. To avoid web performance degradations, service providers need to find the cause of the degradations and improve their web performance. However, web performance depends on web browser running environment such as network environment and calculation resources of clients. In this paper, we propose a platform for web performance measurement which measures not only web performance but also performance of server, network and client with changing web browser running environment. Our experimental result shows that the proposed platform takes 30 minutes to start a measurement of 959 web pages ' performance from 88 hosts, and this does not depend on diversity of web browser running environment. In addition, the platform measures web performance in various web browser running environments by using PlanetLab. As a result, we obtain data for web performance improvement.

Key words Web performance, Measurement platform, QoE, Bottleneck analysis