

Webパフォーマンスの因果分析

中野 雄介^{†,††} 上山 憲昭^{†,††} 塩本 公平^{††}

長谷川 剛^{†††} 村田 正幸[†]

†† 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

††† 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

† 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

E-mail: †{nakano.yuusuke,kamiyama.noriaki,murata}@ist.osaka-u.ac.jp, ††shiomoto.kohei@lab.ntt.co.jp,

†††hasegawa@cmc.osaka-u.ac.jp

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. 本稿では, Webパフォーマンスの低下要因を特定するための, Webパフォーマンスの因果分析の結果を報告する. まず, 我々は様々なWebブラウザ動作環境においてWebパフォーマンスと, Webパフォーマンスに最も影響のある要因を特定するために必要な多様なデータを測定した. その後, 我々はそのような要因を特定するため, 測定されたデータを用いて因果分析を行った. その結果, Webパフォーマンスに最も影響のある要因はWebページの構造とクライアントマシンの性能であることがわかった.

キーワード Webパフォーマンス, 因果分析, QoE

Causality Analysis of Web Performance

Yuusuke NAKANO^{†,††}, Noriaki KAMIYAMA^{†,††}, Kohei SHIOMOTO^{††},

Go HASEGAWA^{†††}, and Masayuki MURATA[†]

†† NTT Network Technology Laboratories, NTT Corporation Midori-cho 3-9-11, Musashino-shi, Tokyo, 180-8585 Japan

††† Cybermedia Center, Osaka University 1-32, Machikaneyama, Toyonaka-shi, Osaka, 560-0043 Japan

† Department of Information Science, Osaka University 1-5, Yamadaoka, Suita-shi, Osaka, 565-0871 Japan

E-mail: †{nakano.yuusuke,kamiyama.noriaki,murata}@ist.osaka-u.ac.jp, ††shiomoto.kohei@lab.ntt.co.jp,

†††hasegawa@cmc.osaka-u.ac.jp

Abstract Web performance has become 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. Web pages with poor performance tend to lose customers. In this paper, we report a result of causality analysis of web performance. First, we measure web performance under various web browser running environments. In addition, we also measure other various data to identify factors that have the most significant impact on web performance. Then, we make a causality analysis to find such factors by using the measured data. The result of analysis shows that the most significant factors are the structure of web page and performance of the client machine.

Key words Web performance, Causality Analysis, QoE

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, Webペー

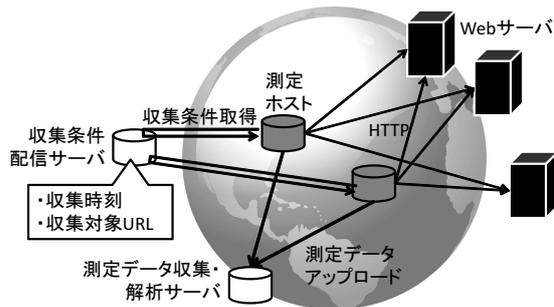


図1 Web パフォーマンス測定プラットフォームの概要

ジの提供者にとって、自身が提供する Web ページのパフォーマンスの良し悪しが重要である。

Web パフォーマンスは Web ブラウザ側の要因と Web ページ側の要因それぞれによって決定される。Web ブラウザ側の要因は、Web ブラウザの種類や動作環境である。具体的には、クライアントの性能や接続されているネットワークの性能などが含まれる。また、Web ページ側の要因は、Web サーバの性能や Web ページの構造である。Web ページの構造は Web ページを構成するオブジェクトの数や内容によって決定される。このような Web ブラウザ側、Web ページ側の多様な要因が影響しあい、最終的な Web パフォーマンスが決定されると考えられる。

このような要因の中で、最も Web パフォーマンスに影響を与える要因を特定することが、ネットワーク事業者にとって重要である。ネットワーク事業者にとって、多くの Web ページのパフォーマンスを底上げすることは、Web ページの提供者・利用者双方に訴求できる。このため、多くの Web ページや Web ブラウザの動作環境において、Web パフォーマンスに大きく影響する要因を特定し、それを改善する仕組みを検討する事が重要である。

そこで本稿では、様々な Web ブラウザの動作環境において、多様な Web ページを表示した際の Web パフォーマンスと、Web パフォーマンスを決定する要因と関連する測定値とを収集し、このような測定値を用いた因果分析により、Web パフォーマンスに影響する要因を特定した。

2. 測定環境

測定環境には、文献[2]の環境を用いた。測定環境について簡単に説明する。

図1に測定環境の概要を示す。測定環境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 [3]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

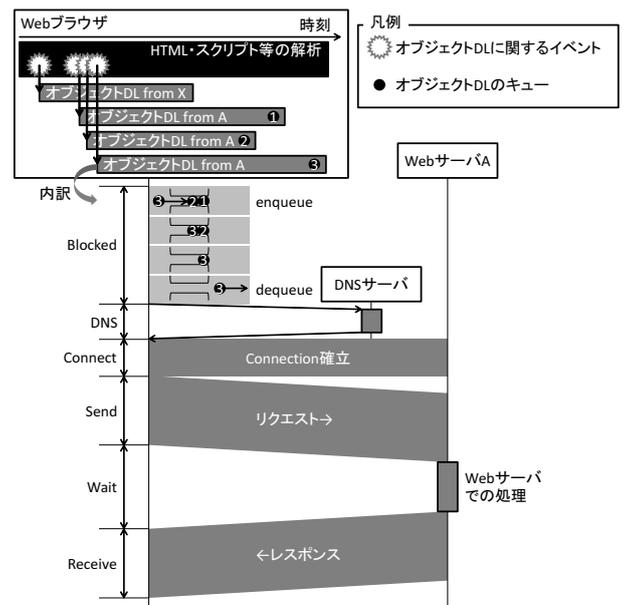


図2 オブジェクトのダウンロードにかかる時間の内訳

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、UnixBench を用いた。

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測定データは 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。

2.1 測定対象

Web パフォーマンスはオブジェクトのダウンロード時間、ブラウザ内での処理時間とそれらの依存関係によって決定される。ここでの依存関係とは、例えば、HTML のパースが進むに従って、HTML に書かれているオブジェクトのダウンロードが開始されることや、JavaScript の処理中は、他のオブジェクトのダウンロードが開始されないことなどが挙げられる。表1に、Web パフォーマンスと、Web パフォーマンスを決定する要因を特定するために測定が必要な測定対象をまとめる。また、それらがダウンロード時間、ブラウザ内での処理時間、依存関係のどれに関連するかを示す。

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、表1の1に示すように、Onload 時間を用いる。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、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

測定対象	説明	ダウンロード時間	処理時間	依存関係
1. Onload 時間	各 Web ページの Onload 時間	✓	✓	✓
2. 静的オブジェクト数	各 Web ページを構成するオブジェクトのうち、閲覧時刻や閲覧場所のような状況によって変化しないオブジェクトの数		✓	✓
3. オブジェクト数	各 Web ページを構成するオブジェクト数		✓	✓
4. オブジェクトサイズ	各 Web ページを構成するオブジェクトのサイズの平均	✓	✓	
5. ホスト数	各 Web ページを構成するオブジェクトを配信するホスト数	✓		✓
6. 依存関係数	各 Web ページに含まれる依存関係の数	✓	✓	✓
7. RTT	各 Web ページを構成するオブジェクトを配信する Web サーバとクライアントとの RTT の平均	✓		✓
8. Blocked	各 Web ページを構成するオブジェクトの Blocked 時間の平均	✓		✓
9. DNS	各 Web ページを構成するオブジェクトの DNS 時間の平均	✓		✓
10. Connect	各 Web ページを構成するオブジェクトの Connect 時間の平均	✓		✓
11. Send	各 Web ページを構成するオブジェクトの Send 時間の平均	✓		✓
12. Wait	各 Web ページを構成するオブジェクトの Wait 時間の平均	✓		✓
13. Receive	各 Web ページを構成するオブジェクトの Receive 時間の平均	✓		✓
14. ベンチマークスコア	Unix ベンチで測定されたベンチマークスコアの逆数 (このスコアが高いほど性能が悪いことを示す)	✓	✓	✓

表 1 測定対象

わけではない。

加えて、Web パフォーマンスを決定する要因を特定するためのデータも収集する必要がある。

表 1 の 6 に示す依存関係数は、先にも述べたようにオブジェクトのダウンロードや処理の間の依存関係である。このような依存関係のカウントには、Web ページを構成するオブジェクトの詳細な解析や、ブラウザの処理中に生成されるイベントのログの解析などが必要となるが、今回は簡易にカウントするため、オブジェクトのダウンロード開始時刻が大幅に変化する回数をカウントした。これは、同時刻にダウンロード開始されたオブジェクトのダウンロードや処理の間には依存関係はないと考えられ、依存関係があるのは、同時刻にダウンロード出来ないオブジェクトの間であると考えられるためである。

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。図 2 は分離された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 2 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。サーバ X に対するオブジェクトのダウンロード開始時刻と、A に対するオブジェクトのダウンロード開始時刻の間に差がある。先に述べたが、このような箇所を依存関係としてカウントする。

表 1 の 8~13 について説明する。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。図

2 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。この時間は Blocked と呼ばれる。なお、Blocked の原因には様々なものがあり、文献 [4] 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。この時間は DNS と呼ばれ、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。これらの時間はそれぞれ Connect, Send, Wait, Receive と呼ばれる。なお、これらは近年の Web ブラウザの機能によって、Onload 時間とともに HAR [5] 形式で保存できる。

また、オブジェクトのダウンロード時間は、RTT に影響を受けると考えられるため、表 1 の 7 に示すとおり、各 Web サーバとクライアント間の RTT も測定する。

2.2 測定条件

今回の測定における条件をまとめる。

使用する Web ブラウザは、Firefox 2.0.0.19 である。これは、PlanetLab のホストが最新の Firefox に対応していなかったためである。また、Firefox の設定としては、オブジェクトをローカルにキャッシュしない、DNS をキャッシュする、Flash・JavaScript を実行する設定にしている。

また、測定地点は 80 箇所であり、地域の内訳は以下のとおりである。これらの測定地点は、PlanetLab で本測定で利用可能であった全てのノードである。

num of obj	California	Wisconsin	...	Japan	average
obj size					
google	600	540		550	550
amazon	700	690		680	690
yahoo	450	400		400	410
:	:	:		:	:
wikipedia	120	120		120	120
average	520	500		510	

図 3 測定データ

North America	55
South America	4
Europe	8
Oceania	5
Asia	8

一方、対象 Web ページは 959 個とした。これは、Web ページのランキングを公開している Alexa [6] の各カテゴリ (ショッピングやニュースなど) ごとのランキング上位 70 の Web ページを抽出した結果である (カテゴリ間で重複している Web ページは 1 つとしているため)。このようにして多様なカテゴリの Web ページを測定対象としている。

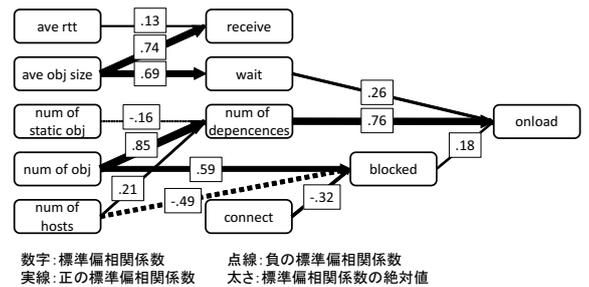
3. グラフィカルモデリングによる Web パフォーマンスの因果分析

Web パフォーマンスの因果分析では、Web パフォーマンスに影響する Web ブラウザ側の要因と Web ページ側の要因それぞれを特定する。つまり、目的変数は Web パフォーマンス、説明変数は Web ページ側、Web ブラウザ側それぞれの測定データとし、目的変数に対して大きく影響する説明変数を特定する。

また、Web パフォーマンスは様々な要因が影響しあって決定される。この様な要因間の因果関係についても明らかにし、Web パフォーマンスがどのように決定されるかを全て明らかにすることが、Web パフォーマンスを改善する施策の検討にとって有用である。このため、測定されたデータを用い、グラフィカルモデリングによる因果分析を行った。

グラフィカルモデルとは、ノードが変数、エッジが変数間の関係を表すグラフ構造である。この様なグラフィカルモデルを推論する作業がグラフィカルモデリングと呼ばれている。本稿では、文献 [7] で解説されている共分散選択による独立グラフのモデリングにより、グラフィカルモデリングを実施し、Web パフォーマンスを決定する要因間の因果関係を表す因果グラフを作成した。

Web パフォーマンスに影響する Web ブラウザ側の要因と Web ページ側の要因と、要因間の因果関係を明らかにするため、グラフィカルモデリングに用いる Web ブラウザ視点の測定データと、Web ページ視点の測定データを収集する。これらのデータは図 3 に示すように、各測定値の測定ホスト、Web ページごとの平均値とした。全測定ホストにおいて共通の Web ページにアクセスしているため、測定ホストごとの測定値の平均値は、測定ホストの特徴を表すデータとなる。同様に、全



Web ページ視点のデータによる因果グラフの適合度

χ^2	130.8	GFI	0.951	RMSEA	0.0631
df	54	AGFI	0.918	NFI	0.954
p	0.0%	SRMR	0.049	CFI	0.972
χ/df	2.423	AIC	22.85		

図 4 Web ページ視点のデータによる因果グラフ

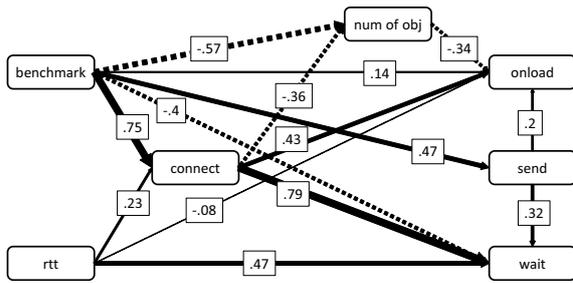
Web ページは共通の測定ホストによって測定されているため、Web ページごとの測定値の平均値は、Web ページの特徴を表すデータとなる。このようなデータを用いてグラフィカルモデリングを行うことで、どのようなホスト、Web ページで Web パフォーマンスが向上するかを特定できる。

3.1 Web ページ視点のデータによる因果分析

Web ページ視点のデータによる因果グラフを図 4 に示す。因果グラフのノードは説明変数を表し、それらを結ぶ矢印は矢印の元から先の説明変数に対する影響があることを示す。また、矢印の太さと矢印に付与されている数字は、その影響の大きさを標準偏回帰係数で表し、実線の矢印は正の標準偏回帰係数を、点線の矢印は負の標準偏回帰係数を表す。なお、因果グラフの下に示す表に因果グラフの適合度を示す。 χ^2 , p 値, χ/df については適合度が低く、実際のデータとモデルとのズレが大きいことを示している。しかし、標準サイズが 1000 以上と大きいため、このような値となっていると考えられる。一方、その他の適合度については概ね高いため、正しくモデリングできていると考えられる。各適合度の値の意味については文献 [7] 参照のこと。

図 4 を参照すると、Onload 時間に直接影響するのは、オブジェクト間の依存関係の数と Wait 時間と Blocked 時間であることが分かる。また、これらの中では依存関係の数が最も影響が大きく、次に Wait 時間の影響が大きいことも分かる。

オブジェクト間の依存関係の要因としては、オブジェクト数、ホスト数の増加、静的オブジェクト数の減少があり、オブジェクト数の増加が最も大きな要因である。オブジェクト数が多いとオブジェクト間の依存関係が増加することは容易に想像できる。一方、静的オブジェクトが多いと、オブジェクト間の依存関係が少なくなる。これは、JavaScript 等による動的なリクエストなどが少ないページは、静的なオブジェクトが多くなり、一方で、JavaScript の処理による待ち時間も短くなるため、オブジェクト間の依存関係が少なくなると考えられる。なお、ホスト数の増加はオブジェクト間の依存関係を増加させるが、一方で、ダウンロードの並列数を増加させ、Blocked の時間を削減する。しかし、Blocked の削減効果が Web パフォーマンス



Web ブラウザの動作環境観点のデータによる因果グラフの適合度

χ^2	0.78	GFI	0.997	RMSEA	0.0
df	4	AGFI	0.980	NFI	0.999
p	94.1%	SRMR	0.011	CFI	1.0
χ/df	0.195	AIC	-7.22		

図 5 Web ブラウザの動作環境観点のデータによる因果グラフ

に与える影響は小さく、ホスト数増大によるオブジェクト間の依存関係の増加が Web パフォーマンスを悪化させる影響のほうが大きく、総合的にはホスト数の増大は Onload 時間を大きくする影響があると考えられる。

Wait 時間はオブジェクトのサイズが大きいと長くなる。これはサイズの大きいオブジェクトを返信する場合、Web サーバ内での処理時間が長くなるためであると考えられる。

以上のように、Web ページ観点で Web パフォーマンスの因果グラフから、Web パフォーマンスに影響を与える要因としては、オブジェクト数やオブジェクトの種類といった Web ページの構造や、Web サーバの性能であることがわかり、主要な要因としては Web ページの構造であることが分かる。一方、RTT のようなネットワーク関連の要因は Web パフォーマンスに影響しないことも分かる。

3.2 Web ブラウザの動作環境観点のデータによる因果分析

Web ブラウザの動作環境観点のデータによる因果グラフと適合度を図 5 に示す。適合度を確認すると、概ね正しくモデリングできていることが分かる。図 5 の因果グラフを参照すると、Onload 時間に直接影響するのは Connect 時間とオブジェクト数と Send 時間とベンチマークスコアであることが分かる。また、これらの中では Connect 時間とオブジェクト数の影響が大きく、Connect 時間が大きくなれば Onload 時間も大きくなる一方、オブジェクト数が増えると Onload 時間は短くなる。このため、Connect 時間が短くなる、または、オブジェクト数が増えるような Web ブラウザの動作環境が Web パフォーマンスの向上に繋がることになる。

なお、オブジェクト数が増えると Onload 時間が短くなることは奇異に感じられるが、オブジェクト数が少なくなるのはオブジェクトのダウンロードがタイムアウトしているためであると考えられる。これは、オブジェクト数が少なくなるのはクライアントの性能が低い場合、または、Connect 時間が長い場合であり、クライアントの性能が低く、オブジェクトのダウンロードがタイムアウトしていることが考えられるためである。

また、Connect 時間が長くなるのはクライアントの性能が低い場合、または、RTT が長い場合であるが、主にクライアン

トの性能が影響している。

以上から、Onload 時間に影響する Connect 時間とオブジェクト数は両方共クライアントの性能に影響を受け、クライアントの性能が低いと Connect 時間は長くなり、オブジェクト数は少なくなり、最終的に Onload 時間が長くなる。一方、RTT はクライアントの性能と比較すると Onload 時間への影響は小さい。つまり、Web ブラウザの動作環境の観点での Web パフォーマンスの因果グラフから、Web パフォーマンスに影響を与える要因はクライアントの性能であることがわかる。一方、RTT のようなネットワーク関連の要因はクライアントの性能と比較すると、Web パフォーマンスに影響しないことも分かる。

4. 考 察

以上の因果分析の結果から、Web パフォーマンスの低下要因は、

- (1) 複雑な Web ページの構造
- (2) Web クライアントの処理性能

であることがわかった。一方、ネットワークに関する要因は Web パフォーマンスにそれほど影響しないことがわかった。

測定ホストの性能が低い場合、ネットワークに関する要因と比較して上記の要因が主要な要因となると考えられる。そこで、今回使用した PlanetLab ホストのベンチマークスコアを確認した。この結果、平均値の性能は、Celeron(E1200) 1.6GHz、メモリ 2GB のデスクトップ PC の性能と同等であることがわかった。このことから、問題となるほど低性能な測定ホストのみを使用したとは言えないと考えられる。

以上から、これまでネットワーク事業者として実施してきた、ネットワークそのものの性能向上や、オブジェクトのキャッシュによる通信の効率化は、最終的な Web パフォーマンスに与える影響は小さいと考えられる。

上記 2 つの Web パフォーマンス低下要因を解決する技術としては、近年、利用が広まりつつあるクラウド連携型ブラウザが有望である。これは、ネットワーク内の装置によって Web ページのレンダリングを行い、その結果を端末上の専用ブラウザが受信し、表示するというものである。スマートフォンでの Web 閲覧など、端末の処理性能が低い状況であっても、クラウド内の潤沢な計算リソースを利用し、高速に Web ページがレンダリングされ、端末内のブラウザはレンダリング結果を表示するだけであるため、上記の要因を解決でき、Web パフォーマンスを大きく改善できると考えられる。

5. 関連研究

Web パフォーマンスに影響を与える要因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 [8]。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 [9] が有名である。なお、Souders は High Performance Web Sites において、RTT が Web パフォー

マンスに影響を与えると述べている。

Web パフォーマンスのボトルネックの診断ツールも開発されている。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 [4]。Wang らは WProf を用い、Web パフォーマンスは主にネットワークの性能によって決まることを特定している。

様々な場所から Web パフォーマンスを測定することで、ボトルネックを特定する研究も行われている。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 [10]。

以上のように、関連研究は Web パフォーマンスの低下要因はネットワークにあると述べている。これは本稿の解析結果と矛盾している。原因としては、これらの文献の実験は特定の環境で実施されたことがあげられる。本稿のような多様な環境での測定結果の解析により、Web パフォーマンスはネットワークの性能ではなく、Web ページの構造やクライアントの処理性能によって決定されることが判明したと言える。

6. おわりに

本稿では、様々な Web ブラウザの動作環境において、多様な Web ページを表示した際の Web パフォーマンスと、Web パフォーマンスを決定する要因と関連する測定値とを収集し、このような測定値を用いた因果分析を行った。その結果、Web パフォーマンスに最も影響のある要因は Web ページの構造とクライアントマシンの性能であることがわかった。

今後は、因果分析の結果をベースにクラウド連携型のブラウザの研究をすすめる。

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] 中野雄介, 上山憲昭, 塩本公平, 長谷川剛, 村田正幸, 宮原秀夫, “Web パフォーマンス測定プラットフォーム,” 電子情報通信学会信学技報 Vol.115 No.159, pp.187–192, 2015.
- [3] “Planetlab”. <https://www.planet-lab.org/>
- [4] 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.
- [5] “Http archive (har) format”. <https://dvcs.w3.org/hg/webperf/raw-file/tip/specs/HAR/Overview.html>
- [6] “Alexa”. <http://www.alexa.com/>
- [7] 矢島隆矢山本将史, Excel で学ぶ共分散構造分析とグラフィカルモデリング—Excel2013/2010/2007 対応版—, オーム社, 2013.
- [8] P. Meenan, “How fast is your website?,” Commun. ACM, vol.56, no.4, pp.49–55, April 2013.
- [9] S. Souders, High Performance Web Sites Essential Knowledge for Front-End Engineers, O'Reilly Media, 2007.
- [10] 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