

Web フォントのサブセット化による Web アプリケーションの性能改善 Performance Improvement of Web Application by Subsetting Web Fonts

森田 孝哉[†] 佐藤 裕介[†] 福士 陽[†]
Takaya Morita Yusuke Satoh Akira Fukushi
清水 理恵子[†] 河合 孝志[‡] 中井 智仁[‡]
Rieko Shimizu Takashi Kawai Tomohito Nakai

1. はじめに

人名や地名には、OS 標準搭載のフォントに収録されていない文字が多数存在する。このため、人名や地名を正確に扱う要件を持つ行政システムでは、システム独自のフォントファイルを作成し、端末へ導入するなどの方法がとられてきた。

近年、行政システムでもクラウドコンピューティングのニーズが高まっており、端末への独自フォントファイル導入を禁止する(端末へのリソース配置を禁止する)技術要件が定義されるようになってきた。

この技術要件を満たす実装方式として、Web フォント方式がある。独立行政法人 情報処理推進機構では 2012 年に Web フォント方式を用いた文字情報基盤実証実験[1]を行っている。

本稿では、文字情報基盤実証実験の成果を活用して、弊社が Web アプリケーションを開発した際に実施した性能改善施策とその適用効果について述べる。

2. Web フォント

2.1 Web フォント方式の概要

Web フォントは CSS3.0 に規定されている技術[2]である。端末に搭載したフォントファイルを用いてブラウザの文字を表示させる従来の方式とは異なり、Web フォント方式ではサーバからダウンロードした Web フォントファイルを用いてブラウザの文字を表示する。Web フォント方式を用いることで端末へのフォントファイル導入を行わずに、ブラウザの文字を表示させることができる。

2.2 サブセット化

日本語のフォントファイルは、収録する文字量の関係から数 MB~数十 MB とファイルサイズが大きくなる傾向にある。そのため、全文字を 1つの Web フォントファイルに収録(フルセット)した場合、Web フォントファイルのダウンロードに時間を要し、ブラウザの画面表示までのレスポンスタイムが悪化する。

Web フォントファイルのサイズに起因するレスポンスタイムの悪化を回避する方法として、Web フォントファイルのサブセット化がある。サブセット化とは、元となる Web フォントファイルを複数の Web フォントファイルに分割することである(図1参照)。

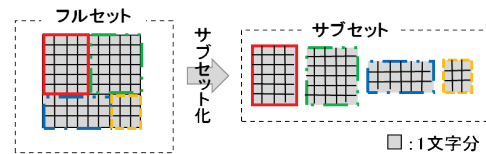


図1 サブセット化の概要

サブセット化により、ブラウザの文字表示に必要な Web フォントファイルだけを選択してダウンロードすることができるようになる。不要な Web フォントファイルをダウンロードすることがないため、レスポンスタイムの改善を図ることができる。

文字情報基盤実証実験でも、Web フォントファイルのサブセット化を行っている。サブセット化の単位は最小の 1文字(1文字サブセット化)である。

3章以降では、行政システムが扱う人名について、最適な Web フォントファイルのサブセット単位を検討し、その適用効果について検証する。

3. サブセット化単位の検討

人名について、Web フォントファイルの適切なサブセット単位を検討する。本稿では国会議員名簿のデータをサンプルとして氏名で扱われる文字の出現頻度を集計し、集計結果を基にサブセット単位を定める。

JIS X 0208:1997 を集計区分として、国会議員名簿[3][4]で使用されている文字について集計した結果を表1に示す。

表1 国会議員名簿の出現文字集計結果

	JIS X 0208:1997			左記以外	合計
	第一水準	第二水準	非漢字		
出現文字数	2702	30	69	43	2844
比率(百分率)	95.2%	1.2%	2.3%	1.4%	100.0%

表1より、国会議員名簿の文字の 95%は JIS 第一水準の文字であり、ほかの区分の文字に比べ突出して出現比率が高いことが確認できる。

上記より、JIS 第一水準の文字集合を 1Web フォントファイルとしてサブセット化し、それ以外の文字については 1文字単位の Web フォントファイルとしてサブセット化することを提案(人名最適サブセット化)する。

4章では 1文字サブセット化と人名最適サブセット化について、実施した評価実験について述べる。

4. 評価実験

4.1 評価実験方法

人名最適サブセット化の適用効果を調べるため、以下の評価実験を実施する。

入力したテキストデータを送信し、次画面で送信したテキストデータを Web フォントで表示する単純な Web アプ

[†] (株) 日立製作所 情報・通信システム社 プロジェクトマネジメント統括推進本部, Hitachi, Ltd. Information & Telecommunication Systems Company, Project & Process Management Division.

[‡] (株) 日立製作所 情報・通信システム社 公共システム事業部, Hitachi, Ltd. Information & Telecommunication Systems Company, Government & Public Corporation Information Svstems Division

リケーションを用いて Web フォントが表示されるまでの時間をレスポンスタイムとして計測する。

評価実験環境として同一セグメント上に Web/AP サーバとレスポンスタイム計測用端末を設置する。また、複数ユーザが同時アクセスした場合のレスポンスタイムについても計測するため、負荷テストツール JMeter を導入した負荷装置も同様に設置する。評価実験環境を図2に示す。

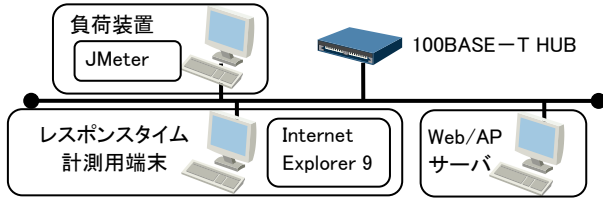


図2 評価実験環境

Web フォントで表示する文字列は、表1に基づいた比率で構成する。評価実験では、Web フォントで表示する文字列の文字数(表示文字数)が100, 250, 1,000, 2,000, 3,000の場合のレスポンスタイムをそれぞれ5回ずつ計測する。また、Web フォントファイルは静的コンテンツであるため、ブラウザがWeb フォントファイルをキャッシュした場合のレスポンスタイムについても同様に計測する。評価実験に使用するWeb フォントファイルは、文字情報基盤実証実験でも使用しているIPAmj明朝フォントファイルを用いて作成する。3章で提案した人名最適サブセット化で作成したサブセットを表2に示す。

表2 人名最適サブセット化で作成したサブセット

	JIS 第一水準文字サブセット	1文字サブセット
収録文字数	4,275	1
ファイルサイズ(eot形式)	629KB	2KB
フォントファイル数	1	55,059

4.2 評価実験結果

評価実験結果を図3, 図4に示す。

Web フォントファイルのブラウザキャッシュ有り・無しそれぞれの場合について、実験結果を述べる。

4.2.1 ブラウザキャッシュ無しの場合

図3より、負荷有り・無しに関わりなく、人名最適サブセット化が、1文字サブセット化よりもレスポンスタイムが常に短くなっている。人名最適サブセット化により、レスポンスタイムが改善している。

また、表示文字数1,000文字で、人名最適サブセット化のレスポンスタイムは1文字サブセット化の場合に比べ、負荷無しの場合で1/7以下、負荷有りの場合で1/10以下の値となっており、レスポンスタイムが改善している。

4.2.2 ブラウザキャッシュ有りの場合

図4より、1文字サブセット化の方が、表示文字数の増加に伴うレスポンスタイムの悪化が大きい。一方、人名最適サブセット化の場合は、表示文字数を増加させても、レスポンスタイムはほぼ横ばいに推移している。負荷有りの場合では表示文字数1,000文字、負荷無しの場合では表示文字数2,000文字に交点があり、1文字サブセット化と人名最適サブセット化でレスポンスタイムの優劣が入れ替わっている。交点位置の表示文字数よりも少ない場合は1文字サブセット化のレスポンスタイムが短く、交点位置の

表示文字数よりも多い場合では、人名最適サブセット化のレスポンスタイムが短くなっている。

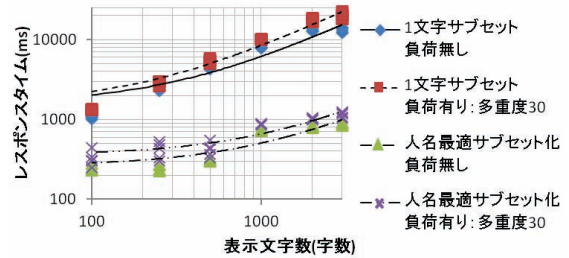


図3 実験結果(ブラウザキャッシュ無し)

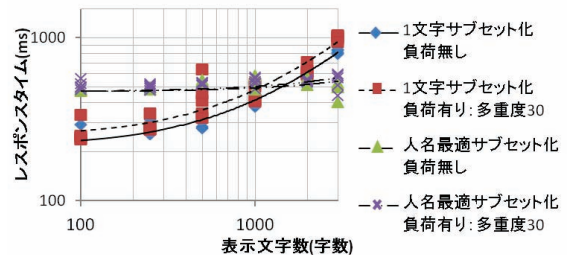


図4 評価結果(ブラウザキャッシュ有り)

5. まとめ

レスポンスタイムの改善を目的に、Web フォントファイルについてサブセット単位を検討した。

ブラウザキャッシュ無しの場合では、人名最適サブセット化を用いることでレスポンスタイムを65%~95%短縮する改善効果が確認できた。

ブラウザキャッシュ有りの場合では、人名最適サブセット化は表示文字数を増加させても、レスポンスタイムが悪化しにくいことが確認できた。また、表示文字数次第では、1文字サブセット化の方が人名最適サブセット化よりもレスポンスタイムに優れていることが確認できた。

6. 今後の課題

ブラウザキャッシュ有りの場合では、表示文字数によっては1文字サブセット化の方が、人名最適サブセット化よりもレスポンスタイムがよくなる場合があるため、サブセット単位の更なる検討が必要であると考える。

また、本稿ではブラウザキャッシュを活用するため、予め生成したWeb フォントファイルを静的コンテンツとして使用した。今後はWeb フォントファイルの動的生成方式[5]についても検討する必要があると考える。

本稿では、人名に最適なサブセット化を検討したが、行政システムが扱う地名についても同様に適切なサブセット化を検討する必要があると考える。

参考文献

- [1]独立行政法人 情報処理推進機構, “文字情報基盤 Web 実証実験 テクニカルレポート.” 2013, (URL : http://mojikiban.ipa.go.jp/wp-content/uploads/2013/06/pdf/Web_Tech_Repo.pdf)
- [2]W3C. CSS Fonts Module Level 3 W3C Working Draft, 2013
- [3]衆議院, “衆議院議員情報.” 2014, (URL : http://www.shugiin.go.jp/internet/index.nsf/html/giin_top.htm)
- [4]参議院, “参議院議員情報.” 2014, (URL : <http://www.sangiin.go.jp/japanese/johol/kousei/giin/186/giin.htm>)
- [5]奥村 命, 案西 稿志, 永井 明彦, 伊藤 孝行, “動的コンテンツに利用可能な日本語 Web フォント配信システム” 情報処理学会 第74回全国大会講演論文集, pp709 - pp711, 2012