美しいフォント高速配信システムの試作と負荷評価

安藤 哲志† 藤田桂英‡ 伊藤 孝行†‡* 永井明彦+

†名古屋工業大学情報工学科 ‡名古屋工業大学大学院産業戦略工学専攻 *MIT スローン経営大学院

+東京工業大学大学院イノベーション専攻

1 はじめに

ウェブ上で様々なフォントを使って，日本人が慣れ親しんだ美しい文字を表示することは難しい．なぜなら，ウェブ上では，ブラウザが存在するクライアントシステムにインストールされているシステムフォントを使わざるを得ないからである．様々なツールが開発されているが，アルファベットに限られたものや，フォントの美しさを犠牲にして画像で表示するなどの技術が多い．本稿では，ユニコードで定義されるほとんどの日本語の，高速配信と表示を可能とし，美しくフォントを表示するプログラムの実装例を示す．また，負荷実験とその評価により，負荷分散の仕組みについても検討する．最後に具体的な応用例についても紹介する．

2 美しい日本語フォント表示の課題

本研究の大きな課題は，ウェブ上で，PDF レベルの美しいフォントで記事を購読したり編集する仕組みを構築することである．元来，ウェブ上に表示される記事は，すべてローカル計算機上に内蔵されているフォントを使って表示される．そのため，美しいフォントで記事を読むというニーズは大変多く，様々な試みがなされている．

PDF は，記事を一つのファイルとして生成し，そのファイルにフォントを埋め込む．そのため，ウェブで表示する場合は，PDF ファイル自体と PDF を解析表示するランタイムプログラムをロードする遅延時間が発生し，ユーザにとって心地よく記事を閲覧することが難しい．

そこで，sIFR[1] というフラッシュをベースにした技術が開発されている．sIFR は，ページの中でフォントを置き換えたい部分だけ，小さなフラッシュファイル（swf ファイル）を表示させる仕組みである．sIFR によって，ウェブ上の文字を任意のフォントで表示することが可能だが，アルファベットのみに限られている．アルファベットのみの場合，フォントの書体ごとのファイルの大きさはそれほど大きくなく，数十キロバイト程度である．従って，sIFR では，フォントを全部ローカルな計算機上のウェブブラウザに送信している．

sIFR のように，フォントの書体ファイル全体をローカルな計算機上に送信する場合，以下のような具体的な課題が明らかになっている．

【日本語フォントの書体の数】欧米のアルファベットは，英語であれば書体は２６個である．記号などを合わせても ASCII 文字コード内に収まる．従って欧米のフォントのファイルは数～数十キロバイトである．一方，日本語の場合，漢字も含まれるため，フォントの書体の数は極めて多い．従って，日本語フォントのファイルは，ユニコードを含まない場合でも，数メガバイトになる場合が多い．数メガバイトのファイルを，ウェブページに配信すると，現状の一般的な帯域において，一般的なユーザにとって大きな遅延が生じる．そこで，本研究で試作したシステムでは，日本語フォントのファイルの中でも必要な文字の書体だけを送信する．

【検索エンジンへのインデクシング】ウェブページ上の文字を，画像ファイルや swf ファイルで置き換えた場合，検索エンジンにインデクシングされない．そこで，HTML の内部に，Alternative な文字として，埋め込んでおく．

【コピーアンドペーストのニーズ】ウェブページの便益の一つとして，内容をコピーアンドペーストできるという点がある．内容のコピーアンドペーストは，現実的なニーズである．書体だけを送信するということであれば，JPEG や GIF 等の画像ファイルとして送付することが考えられる．しかし，画像ファイルをウェブページに表示するだけでは，コピーアンドペーストのニーズに答えることができない．また，画像ファイルとした場合，ビットマップになるため，フォントの書体のアウトラインの定義が有効に利用できないという欠点もある．

【ユニコードへの対応】日本語および，中国語を含むアジア圏のフォントを実現するためにユニコードに対応したフォントが多く出荷されている．ユニコードでカバーされる書体すべてを一つのファイルとして見ると，数メガバイト～数十メガバイトのサイズになる．従って，ユニコードでカバーされるような，多様で多種な書体の送信を実現するには，必要な文字だけを送信する機能が不可欠である．

【知財としての問題】すべての書体が含まれるフォントファイルを直接配信した場合，知財としての問題もある．特に swf などのアウトラインを含むフォーマットの場合，すべての書体を送信してしまうのではなく，その１部分だけを送信することが望ましい．

以上の課題を解決するために，本研究では，日本語フォントのファイルの中で必要な文字の書体だけを送信するプログラムを実装した．

3 フォント高速配信システム

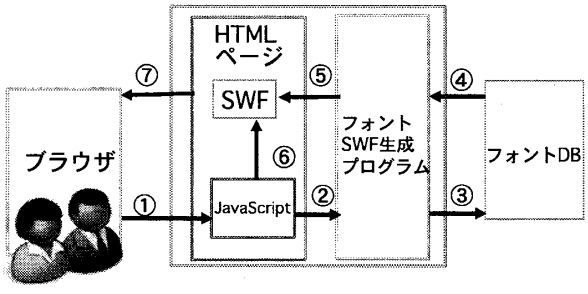

図 1: フォント高速配信システムの概略

図１にフォント高速配信システムを示す．本システムは，フォント DB，フォント SWF 生成プログラム，及び JavaScript のプログラムから構成される．ユーザがブラウザから要求した HTML に JavaScript が埋め込

An Implementation of Beautiful Font Casting System and its Evaluation on Load

†Tetsushi Ando ‡Takayuki Ito

†Dept. of Computer Science, Nagoya Institute of Technology

‡School of Techno-Business Administration, Nagoya Institute of Technology& MIT Sloan School of Management & Tokyo Instiute of Technology

まれており，必要なswfを作成しHTMLに埋め込んでユーザのブラウザに表示する．本システムは，Java，アプリケーションサーバTomcat，及びJavaScriptを用いて実装されている．

図1でのシステムの流れを説明する．①でユーザがブラウザから要求を送る．②でJavaScriptプログラムから必要なフォントの作成する命令がフォントswf作成プログラムに送信される．③でフォントDBに必要なフォント情報を問い合わせ，④でフォント情報がフォントswf作成プログラムに送付される．⑤でフォントswf生成プログラムによってswfが生成される．⑥でJavaScriptプログラムがHTMLページに生成されたswfファイルを埋め込み，⑦でブラウザに表示する．

4 負荷実験と考察

試作システムの実環境での性能を評価するために，表1の環境で負荷実験を行った．

表1: 負荷実験の環境

CPU	Dual Core AMD Opteron 285 2590Mhz x2
メモリ	8GB
OS	openSUSE10.3(64bit)
Java	version 1.6.0
Tomcat	version 5.5

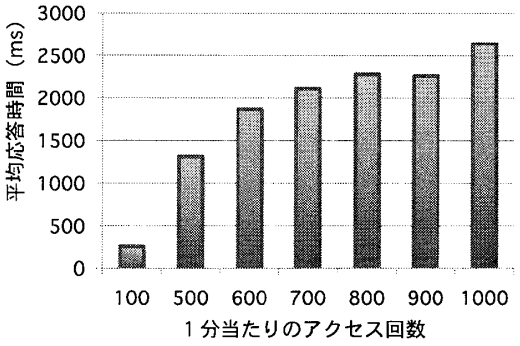

図2: 平均応答時間

図2に，1分当たりに１００回から１０００回アクセスした場合の（アクセスが成功した場合の）平均応答時間を示す．アクセス成功の割合が高い，1分当たり５００回程度の場合およそ，１．２秒かかっている．

図3に，1分当たりに１００回から１０００回アクセスした場合のアクセス成功とアクセス失敗の割合を示す．1分当たり５００回アクセス程度までは，すべてアクセスに成功するが，1分当たり６００回アクセス程度からアクセスの失敗が観察された．理由はいくつか考えられるが，ヒープメモリの大きさではないことが分かっている．図2で示したように，swfを生成するのに，1秒から2秒かかっている．そのため1秒に１０アクセス以上あると処理が追いついていない可能性がある．

実用上は当初は10,000ユーザ程度の使用を想定している．平均的には1秒間の同時アクセス数はそれほど増加しないと予想しており，サーバを複数台用意するなど分散処理を行うことで対応可能と考えている．現在，処理の高速化の工夫とキャッシュの導入を検討中である．

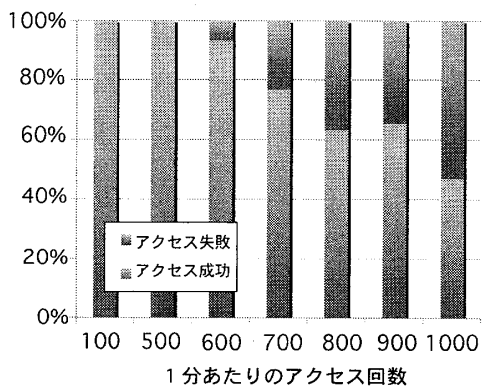

図3: アクセス成功の割合

5 応用例

図4は，本フォント配信システムのブログシステム（試作）への応用例である．図4に示すように，多様なフォントでブログの記事を記述することができることが特徴である．また，文字列のコピーアンドペーストも可能である．

図4: ブログシステムへの応用例

6 まとめ

本稿では，ユニコードで定義されるほとんどの日本語の，高速配信と表示を可能とし，美しくフォントを表示するプログラムの実装例を示した．また，負荷実験とその評価により，負荷分散の仕組みについても検討し，最後に具体的な応用例を紹介した．アジア圏の国々のめざましいITの発展を考慮すると，アジアの多様で多種な言語の美しいフォントを送信するニーズはますます増えることが期待できる．本研究で提案した手法は，あらゆるフォントに対応できる手法である．

参考文献

[1] sIFR, http://wiki.novemberborn.net/sifr/, 2008.