

閲覧性の維持と通信量の削減を両立した ホームページの作成事例

キーワード：ホームページ、表示高速化

はじめに

光回線などのブロードバンド網の整備に加え、携帯電話回線などを用いたモバイル環境の普及も急速に進んでおり、様々な場所からインターネットに接続できるようになりました。またスマートフォンやタブレットなどの普及により、どこでも手軽にホームページを閲覧できる環境が整いつつあります。その一方で、利用者の急激な増加に伴い、人口密集地や特定の時間帯においてインターネット回線が混雑し、Web 閲覧の応答性能低下やタイムアウトの発生といった問題が頻発するようになりました。また、MVNO（仮想移動体通信事業者）の台頭や通信キャリアの定額料金制の廃止などから、個々の利用状況に適したサービスを選ぶとともに、通信量の「節約」を求める利用者が増えつつあります。

本記事では産技研ホームページにおける、閲覧性（表示品質）の維持と通信量（データ量）の削減を両立したコンテンツの作成事例についてご紹介いたします。

ホームページ作成を取り巻く現在の状況

現在、ホームページを作成するためには、HTML、CSS、JavaScript の3つの記述言語を組み合わせる方法が一般的です。それらに画像、動画ファイル等が添付されます。

また、より高度なウェブアプリケーションを構築可能とするため W3C¹⁾ より HTML5 が 2014 年 10 月に、CSS3 の主要な仕様が 2011 年～2012 年にかけて勧告されました。これにより、HTML・CSS・JavaScript の組み合わせだけでブラウザ上で多彩な表現が実現出来るようになりました。しかし、利用者の環境によっては、HTML5/CSS3 に対応できないケースが多く、ホームページ開発者は HTML5/CSS3 対応ブラウザの利用者向けに

高品質なホームページコンテンツを提供する一方で、HTML5/CSS3 非対応ブラウザの利用者に配慮したホームページコンテンツを提供する努力が求められています。

通信量削減の実例

1. CSS を用いたページ作成

これまでの CSS は、デザインの自由度が少なく、また、同じ指定でもブラウザによって表示に差異が生じることがありました。このため、ブラウザ間の互換性を考慮し、「文字」であっても画像を用いるケースが多く見受けられました。しかし、現在はブラウザの大半が少なくとも CSS2.1 までサポートするようになりました。CSS2.1 の機能を活用することで、これまで画像ファイルで表現していた選択ボタンなどを、CSS と HTML だけで表現できるようになり、通信回数やデータ量の削減による応答時間の短縮が期待できます。

2. 画像の圧縮

画像データ量削減の取り組みとして、人が気づかないレベルで圧縮を行う方法があります。一例として pngquant コマンドをご紹介します。pngquant は BSDLicense²⁾ の下、無償で使用が可能で、PNG 形式の画像を 256 色以下に減色しデータ量を削減します。図 1、図 2 に産技研のロゴマークを 16 色に不可逆圧縮した場合の結果を示します。



図 1. 元画像（24 ビットカラー、12KB）



図 2. 16 色に減色した画像（3.5KB）

この例では、解像度を下げずにデータ量を約1/3程度に削減することができました。このように減色による不可逆圧縮は、色数の少ないロゴマーク等で特に効果を発揮します。

3. コンテンツの minify 化

コンテンツ制作時は、各種記述文の中に、適度なインデントやコメント、空白行などを挿入して可読性を高めます。しかし、これらは Web ページ生成データとしては余分な情報です。これらのデータを取り除くことで通信量の削減が可能になります。この手法を minify 化と呼びます。手軽に minify 化の効果を確認するツールの一例として、Google Chrome 用の拡張機能である PageSpeedInsights をご紹介します。これは ApacheLicense2³⁾ の下、無償で利用が可能です。minify 化を行った後のコンテンツは、図 5 のとおり可読性が低下するため、管理のための編集作業は minify 化前(図 4)の状態で行い、サーバに配置する時に minify 化(図 5)する、といった運用を推奨します。

```
@charset 'UTF-8';
/*--body section--*/
div#body{
  background-color: #F7E9F7;
  overflow: hidden;
}
div#body h2{
  color: #E03B7A;
  background-color: #fff;
  width: 4em;
  margin: -.2em;
  margin-top: -1.5em;
  margin-bottom: .2em;
```

図 4. minify 化前の CSS 例(5.8KB)

```
@charset 'UTF-8';div#body{background-color:#F7E9F7;overflow:hidden;}div#body h2{color:#E03B7A;background-color:#fff;width:4em;margin:-.2em;margin-top:-1.5em;margin-bottom:.2em;transform:rotate(-15deg);border-radius:4px;border:3px solid #C7A1E3;text-align:center;position:relative;text-shadow:-2px -2px 1px #fff,-2px 2px 1px #fff,2px -2px 1px #fff,2px 2px 1px #fff}div#body h2 > a{display:block;margin-top:-.6em;overflow-
```

図 5. minify 化後の CSS 例 (4.5KB)

4. CDN の活用

CDN (コンテンツ・デリバリ・ネットワーク) は JavaScript ライブラリなどの外部読み

込みファイルを高速に配信するサーバ群です。JavaScript ライブラリの一つである jQuery を例に挙げると、図 6 のように参照ライブラリを自サイトから CDN へ変更することで、自サーバへの通信回数を削減できます。また、クライアントにおいても、CDN が配信するファイルをブラウザがキャッシュすることで、同じファイルが必要なサイトを閲覧する際に、通信量の削減が期待できます。その反面、CDN が配信するファイルに悪意あるコードが埋め込まれた場合、自サイトの利用者が被害を受ける可能性があり、また、CDN に障害が発生した場合は、ホームページが正常に表示されなくなる、といったリスクもあります。このため、CDN を用いる場合、信頼のおける配信元を選択する必要があります。

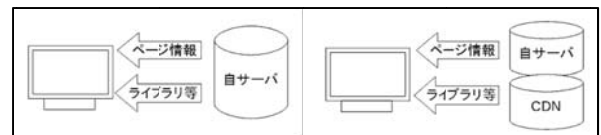


図 6. jQuery 等のライブラリ配信を CDN に切り替えた場合のイメージ

まとめ

サーバの応答性能を維持するための効果的な方法として、nginx など動作が高速なウェブサーバソフトウェアの導入や、gz 圧縮の導入による通信量の削減、といったサーバ構成の変更に基づく対応が挙げられます。これらサーバでの対応に比べ、本事例の効果は小さなものです。しかし、サーバをレンタル利用している場合、サーバ構成には関与できません。また、自社で運用しているサーバの場合でも、構成変更を行う際には、設定作業や安全性の再確認など様々なコストが発生します。本稿では、少ないコストで手軽に試行・実施できる事例を紹介しました。これらの事例が企業における情報発信の一助となれば幸いです。

1) <http://www.w3.org/>

2) <http://opensource.org/licenses/bsd-license.php>

3) <http://www.apache.org/licenses/LICENSE-2.0>