

頻出文字列の共有による Web 文書の転送容量削減について

上平 竜也[†] 南角 茂樹[†]
[†] 大阪電気通信大学大学院総合情報学研究所

1. はじめに

デジタルメディアの普及により多くの人が Web サイトを利用するようになり、ネットワークのトラフィックは増加している。また、スマートフォンの普及とそれに伴う通信サービスの多様化により、遅い回線を利用するユーザーも増えてきている。そのため容量の大きい Web サイトにアクセスすると表示に遅延が発生し、ユーザーエクスペリエンスに悪影響を与えてしまう。快適に Web サイトを利用してもらうためにはデータの転送容量を減らし、遅延を小さくする必要がある。

多くの Web サイトでは各ページで同一のテンプレートや要素を用いている場合が多いが、それぞれのページにアクセスする度にそれらを毎回サーバーから転送して読み込んでいる。そこでそれらを一度読み込んだ後は再利用することで、転送容量を減らすことが出来るのではないかと考えた。

そこで頻出要素の共有による転送容量の削減について提案する。本稿では転送容量を減らし表示の遅延を小さくすることを目的とする。

2. 提案手法

2.1 概要

提案システムの概要を図 1 に示す。サーバーは各ページに頻出する要素から共有資源を作成し、レスポンス送信時は内容から共有資源と同一の文字列を切り取り、共有資源と共に送信する。クライアント側は共有資源を用いて元の内容を復元し表示する。同じ共有資源を用いるページにアクセスする際、共有資源を要求する必要がないため転送容量を削減することができる。

2.1 サーバー側

最初にサーバー側で共有資源を作成する。クライアントからリクエストが送られてきた際、レスポンス内容をデータベースに蓄積していく。一定周期でデータベースの内容から頻出する要素を探し出し、共有資源を作成する。

レスポンス内容に共有資源と同一の文字列が含まれている場合、その文字列を共有資源があったことをクライアントに伝えるための情報に置き換えてから送信する。その際クライアントが共有資源を保持していない場合、同時に送信する。

2.2 クライアント側

サーバーへリクエストを送る際に、クライアントの共有資源の保持情報を同時に送信する。サーバーからのレスポンスは共有資源部分が置換された状態なので、ブラウザ表示前にサーバーでの置換前のデータに戻してから表示する。

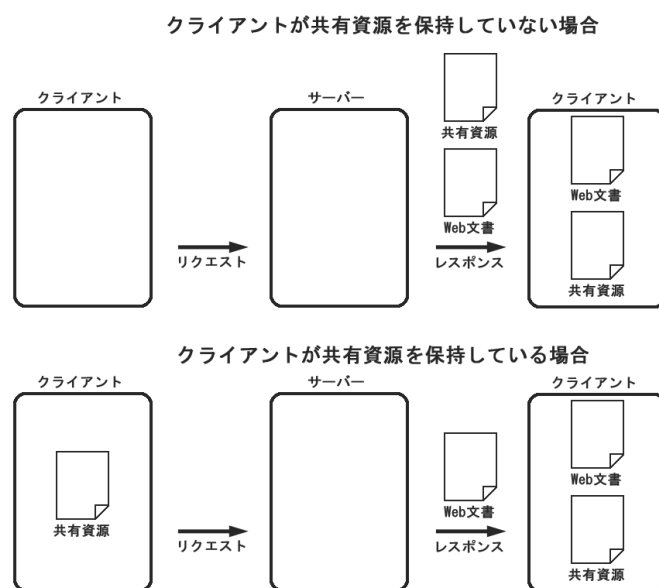


図1. 提案システムの概要

4. 実装

サーバー側処理は Web サーバで最も使われている Apache 上に実装する。Apache はモジュール構造を採用しており、レスポンス処理に割り込んで機能を拡張することができる。クライアント側処理は Javascript を用いて実装する。

5. まとめ

本稿では頻出要素を共有することによる HTML 文書の転送容量削減についての提案を行った。今後の予定として、提案システムの実装を行い、既存のシステムとの比較評価を行う。

参考文献

[1] 柴田 和祈:「動的 Web アプリケーションのためのプリフェッチ機構」情報処理学会論文誌 Vol54 No.2, pp647-658(Feb. 2013)