

# ウェブブラウザの画面及び操作の同期方式の検討

加島 隆博 羽藤 淳平  
三菱電機株式会社 情報技術総合研究所

## 1. はじめに

ウェブ技術の発達とウェブブラウザ搭載機器の普及により、実用性の高いウェブアプリケーションを、どこでも利用することが可能になった。そこで我々は、複数の端末で動作している各々のウェブブラウザの表示内容を同じにする画面の同期と、どこからでも同じ操作を可能にする操作の同期に着目した。これにより、例えば複数人による共同文章作成のような利点が考えられる。

従来このような画面及び操作の同期を実現するには、通信量が多いという問題があった。そこで我々はウェブブラウザの特徴を活かして、通信量を削減しつつ、ウェブブラウザの表示内容を忠実に同期する手法を検討した。

本論文では、まず従来の同期手法の問題点を挙げ、次に提案手法を論じ、実験プログラムによる同期結果と通信量を示す。

## 2. 従来の同期手法

複数の端末で動作するウェブブラウザの表示内容をリアルタイムで同一にするには、ある1台の端末の表示内容を画像データとして他の端末に転送することで実現できる。この手法を用いるソフトウェアとしてVNC [1]がある。VNCの目的は端末の遠隔操作であるが、ウェブブラウザの画面及び操作を同期するという用途でも使用できる。この場合、1台の端末をサーバとし、他の端末をクライアントとする。サーバの利用者は通常通りサーバ自身のウェブブラウザを操作し、クライアントの利用者はVNCを使用して、サーバのウェブブラウザの画面を表示及び操作することで、画面及び操作の同期を実現できる。

しかし、VNCの課題として通信量が多いことが挙げられる。例えば、画面の画素数が640×480で、画素ごとに24ビットを使用する環境の場合、画像1枚はおおよそ1MBのデータになる。これを1秒間に30枚転送したい場合、200Mbps以上の通信回線が必要である。この通信量は現実的ではないため、画面内の変更があった領域のみ転送することや、転送

データの圧縮を行うことにより、通信量を削減しなければならない。

そのため、我々はウェブブラウザの特徴を活かして、通信量を削減可能な画面同期システムについて検討した。次章で提案手法について述べる。

## 3. 提案手法

ウェブページは1つのDOMツリーで内部表現され、ウェブブラウザはそれを元に描画する。そのため、全てのウェブブラウザのDOMツリーを、同一のDOMツリーに同期することで、全てのウェブブラウザの表示内容を同一にできると考える。また、各ノードの内容は、タグ名、属性、親子関係等の単純なデータであるため、VNCのように画像を転送するよりもデータ量を削減できると考える。ノードに変更があった際も、そのノードの変更点のみを伝えれば良い。更に、ページに画像等の外部リソースが用いられている場合、これらもDOMツリーとは別に同期を行うことで、リソースも利用可能になる。

次章以降で、提案手法の実験プログラムと実験結果について述べる。

## 4. 実験プログラム

本章では実験の目的と、実験プログラムの詳細について述べる。

### 4.1. 目的

我々の研究の目標は、ウェブブラウザのDOMツリーの同期による、画面及び操作の同期であるが、今回はその研究段階の第一歩として、以下の点を明らかにするためにプログラムを用いて評価を行った。

1. DOMツリーとリソースの同期により、画面内容を同一にすることが可能か。
2. VNCのような画面の画像を同期する手法に比べて、通信量を削減することは可能か。

### 4.2. 構成

実験プログラムの構成は、1台のウェブブラウザをサーバとし、その他のウェブブラウザをクライアントとした。また、サーバとクライアントはJSONによるデータ記述法を用いて通信を行うこととした。

サーバはページ読込開始時とページ読込完了時に、クライアントに通信を行う。それぞれの処理内容と通信内容を次節以降で詳説する。

### 4.3. ページ読込開始時

サーバでウェブページの読み込みを開始した場合、図 1 の JSON をクライアントに送信する。

```
{"type": "Initialize"}
```

図 1 ページ読込開始時に送信する JSON

クライアントはこの JSON を受信した場合、Document ノードの子ノードを全て削除する。

### 4.4. ページ読込完了時

サーバでページの読み込みを完了、すなわちウェブブラウザによって DOM ツリーの構築が完了した際に、サーバは以下の処理を順に行う。

1. 各 DOM ノードに対して一意の ID を設定
2. 各 DOM ノードの情報をクライアントに送信
  1. の ID の設定は、サーバとクライアント間で、各ノードを一意に識別できるようにするために行う。
  2. のクライアントへの送信は、トップダウンの順で Document ノードの子ノードの情報をクライアントに送信する。送信対象は Element ノード及び Text ノードのみとする。Element ノードの場合、図 2 の JSON を送信する。

```
{
  "type": "InsertNode",
  "data": {
    "nodeType": 1,
    "parentNodeID": (親ノードのID),
    "tagName": (タグ名),
    "attributes": { (…属性一覧… ) },
    "style": { (…スタイル一覧… ) }
  }
}
```

図 2 ノードごとに送信する JSON

Text ノードの場合は、Element ノードと異なり tagName 等を含まず、テキストを表す textContent 属性を送信する。Element ノードと Text ノードは、nodeType の値によって識別する。クライアントはこれらの JSON を受け取り次第、JSON を元にノードを作成して自身の DOM ツリーを構築する。

## 5. 実験結果

本章では実験プログラムによる同期結果と通信量について示す。なお、今回の実験で使用した同期対象のウェブページは、①文字が多いページ (<http://ja.wikipedia.org/>) と、②画像が多いページ (<http://www.mitsubishielectric.co.jp/>) である。

### 5.1. 同期結果

提案手法の DOM ツリーとリソースの同期により、サーバとクライアントの両方のウェブブラウザで、同じ画面表示を得ることができた。しかしながら①のページでは、JavaScript の document.write が使

用されており、クライアントの DOM ツリーの構築中にこのスクリプトが評価されてしまったことで、正常に同期できなかった。そのため、このスクリプトを動作しないようにする必要があった。

### 5.2. 通信量

VNC に代表される画面の画像を同期する手法(以下、画像方式)の通信量と比較するために、ウェブブラウザのスクリーンショットを PNG 形式にしたデータ量と比較した。通信量の結果を表 1 に示す。

表 1 通信量

	DOM	リソース	合計	画像
①	330	226	556	1,139
②	92	1,138	1,230	519

(単位: KB)

DOM 列は DOM ツリーの同期に要した通信量を、リソース列はウェブページで用いられている画像等の外部リソースの取得に要した通信量を、合計列は DOM 列とリソース列の合計で、提案手法の通信量の合計を示す。画像列は画像方式のデータ量を示す。

表 1 を見てわかるように、①においては画像方式に比べて提案手法の通信量を半分に抑えることができた。②においては逆に、提案手法の方が画像方式よりも通信量が倍以上に上回った。これはリソース列の値が示すように、ウェブページで使用されている画像が多いためである。但し、②のページではスクリプトを用いて時分割で画像を切り替える演出を行っていた。そのため、表 1 における提案手法の値は、ウェブページで使用される全ての画像のデータ量を含むが、画像方式ではスクリーンショットが撮られた瞬間の画像しか含まない。スクリプトによって画像が切り替わった際に、画像方式では変更領域の画像転送が必要であるが、提案手法では既にリソースを持っているため追加の画像転送は必要ない。

## 6. まとめ

本稿は DOM ツリーを同期することによる、ウェブブラウザの画面同期について提案し、実験プログラムによって同期結果と通信量を示した。DOM ツリーを同期することにより、画面の同期が可能なことと、画像を転送するよりも少ない通信量で、同じ表示が得られることを示した。

今後はサーバ及びクライアント双方による、操作の同期の実現を目指して研究を行う予定である。

## 参考文献

- [1] T. Richardson et al., "Virtual network computing," *Internet Computing, IEEE*, vol. 2, no. 1, pp. 33-38, Jan/Feb 1998.