

# ブラウザのキャッシュと利用者操作ログを用いた Web アプリケーション高可用化手法の実装と評価

荒井 大輔 吉原 貴仁  
株式会社 KDDI 研究所

## 1. はじめに

近年、その手軽さから Web ブラウザのみで利用者にメール機能などを提供する Web アプリケーション(以降、Web アプリと呼ぶ)が急速に普及しつつある。Web アプリの多くは、その利用にネットワーク接続が必要である。このため、ネットワーク接続が正常でないと Web アプリの利用が利用不可となり、その可用性が低下するという課題がある。この課題に対し著者らは、Web ブラウザの全通信をキャッシュする機能と、マウス操作等の利用者操作ログを取得する機能を Web ブラウザに追加することで、オフライン中も Web アプリの利用を可能とする手法を提案している[1]。本稿では、提案手法の実装概要とその評価について述べる。

## 2. ブラウザのキャッシュと利用者操作ログを用いた Web アプリ高可用化手法[1]

### 2.1. 想定環境

想定環境を図1に示す。Ajax(Asynchronous JavaScript + XML)[2]等により実装された Web アプリの利用を想定する。利用者は利用者 PC 内の Web ブラウザより、Web サーバに HTTP 接続する(図1下)。Web ブラウザは、利用者の URL 入力等に従い、Web サーバより HTML や画像、JavaScript ファイル等を取得し(図1(R1))、取得内容を表示領域に出力する(図1(R2))、例えば、メールのメイン画面を表示)。利用者は表示内容に対し、キーボードやマウス操作を行う。利用者の操作は JavaScript が取得し(図1(R3))、例えば、着信メールを選択)、XMLHttpRequest を用いて、表示領域の遷移とは非同期に、Web サーバに送信、応答を受信する(図1(R4))、例えば、メールアドレスを受信)。JavaScript はサーバの応答に従い、表示データを組立て、表示領域を変更する(図1(R5))、例えば、メール本文を表示)。ネットワーク接続はネットワーク障害や、地下鉄乗車による通信圏外等によりオフラインとなる。

### 2.2. Web アプリ高可用化手法[1]の概要

本手法では、Web ブラウザと Web サーバ間の全ての通信を Web ブラウザ上で監視し内容を保存するキャッシュ DB(図1(DB1))と、オフライン中の利用者による操作履歴を管理する利用者操作ログ管理 DB(図1(DB2))、メール作成などオフライン中の利用者操作をオンラインとなった後に Web サーバに反映するオフライン中操作反映機能(図1(F1))を利用者操作 PC に新たに追加する。各 DB と機能は、次の3つの手順により Web アプリを高可用化する。提案手法[1]の処理手順を図1とともに示す。

#### (手順1)全通信の監視と保存

キャッシュ DB(図1(DB1))は Web ブラウザの全通信を監視し、オンライン中、全ての送受信内容(図1(S1)、(S2))を保存する。

#### (手順2)キャッシュによる応答と利用者操作ログの保存

Web ブラウザからの要求に対するタイムアウト等を検知した場合、オフラインと判断し、過去の応答から、要求をキーとして検索し、検索結果を応答する(図1(S3))。

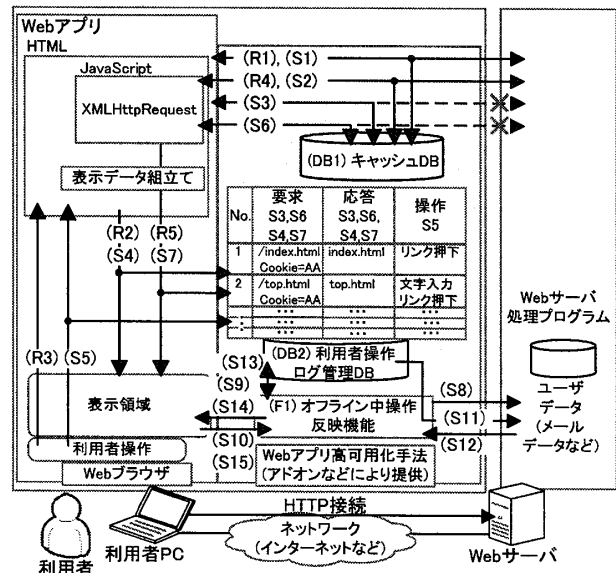


図1: 想定環境と Web アプリ高可用化手法[1]の概要

Web サーバへの要求や応答に日付や乱数を付与する Web アプリに対応するため、検索にはワイルドカード検索を用いる。検索結果が複数となった場合、利用者に検索結果の一覧をサムネイル表示し、一覧から選択可能とする HTML を応答する。キャッシュ DB の応答は Web ブラウザにより表示領域に出力される(図1(S4))。オフライン中の利用者操作は、Web サーバに代わりキャッシュ DB が応答する点を除き、通常の Web アプリと同様となる(図1(S5)、(S6)、(S7))。オフライン中、要求と応答(図1(S3)、(S6))、表示領域の内容(図1(S4)、(S7))、利用者操作(図1(S5))を関連付け、実行順を示すシーケンス番号と共に利用者操作ログ管理 DB(図1(DB2))に保存する。

#### (手順3)利用者操作ログの Web サーバへの反映

オフライン中操作反映機能(図1(F1))は、利用者操作ログ管理 DB にエン트리がある場合、定期的に Web サーバへの接続を確認する(図1(S8))。Web サーバへの接続が確認された場合、利用者に Web サーバへの反映が未了のオフライン中操作があることを通知し(図1(S9))、利用者はオフライン中の操作の反映を要求する(図1(S10))。オフライン中操作反映機能は、利用者操作ログ管理 DB よりエン트리毎に読み込み、サーバへの反映を行う(図1(S11))、(S12)、(S13)、(S14)、(S15))。サーバへの反映の際、例えばオフライン中に Cookie が期限切れとなり、同じ要求に対する応答が、キャッシュ DB からは Web アプリにログイン済みの画面、サーバからはログイン要求画面となるなど、異なる場合がある。その場合、反映を一時中断し、例えば Cookie の再取得のため、利用者に入力を求める。

## 3. Web アプリ高可用化手法の実装

提案手法の評価のため、Windows XP SP3 上で動作する Internet Explorer 8 用のブラウザアドオンとして、Web アプリ高可用化手法[1]を実装する。

実装システムの概要を図2とともに示す。実装システムは、Webブラウザからの要求や、それに対する応答を中継するプロキシのように振る舞う。また、各DBは、SQLite 3.1[3]により、アドオンからアクセス可能なデータベースを利用者PC上に作成することで実装する。

実装システムは、以降に詳述する5つのIE(Internet Explorer)オブジェクトから構成される。5つのIEオブジェクトのうち、同時に作成、実行するIEオブジェクトは、ネットワーク接続の状況や、(手順1)~(手順3)のいずれを実行するかで異なり、不要なIEオブジェクトは都度削除することで、利用者PCの計算機資源を開放する。これは、Windows Mobileが動作するモバイル端末のような計算機資源の限られた環境での動作を想定している。

#### (F1)メイン処理用IEオブジェクト

常時実行される。Webアプリ高可用性手法のメイン処理を実行するためのIEオブジェクト。Webブラウザが利用者PC上で起動されると同時に作成され、Webブラウザ起動中は常時実行される。Webブラウザからの要求を全て中継し、送受信情報をキャッシュDBに保存する。また、ネットワーク接続の状況や各手順の実行を管理し、IEオブジェクトの作成と削除を行う。さらに、(F1-1)利用者操作ログ取得用JavaScript埋め込み機能と(F1-2)スナップショット取得機能を持つ。(F1-1)は、サーバ応答をWebブラウザに返す前に、利用者操作ログを取得するため、HTMLのBODYエレメントに、イベントをフックし実装システムに通知するJavaScriptを埋め込む。(F1-2)は、フレームワークが提供するOnDocumentCompleteイベントにより、実装システムは画面表示の完了を取得し、(手順2)において検索結果をサムネイル表示するため、画面のスナップショットを取得する。

#### (F2)Webクローラ用IEオブジェクト

オンライン中のみ実行される。利用者がWebサーバより取得したHTMLファイルを解析し、利用者からの要求とは別に、リンク先にHTTP接続し、受信データをキャッシュDBに保存する。これにより、例えば利用者がオンライン中にメーラのメイン画面を表示するのみで、オフライン中の未読メールの閲覧が可能となる。

#### (F3)DOM構造取得用IEオブジェクト

オフライン中のみ実行される。(手順3)において、サーバとキャッシュDBからの応答を比較するため、キャッシュDBからの応答内容のDOM(Document Object Model)構造を取得し、利用者操作ログとともに利用者操作ログ管理DBに保存する。また、DOM構造を比較用とするため、サーバへのアクセス毎に変更の可能性があるバナー広告などをフィルタする機能を有する(F3-1)。

#### (F4)オフライン中操作反映用IEオブジェクト

オンライン中操作反映時のみ実行される。利用者操作ログ管理DBから、利用者操作ログを読み込み、利用者に代わりバックグラウンドでオフライン中の利用者操作をWebサーバに反映する。サーバ応答とキャッシュDBからの応答が異なる場合、(F1)メイン処理用IEオブジェクトを通じて、利用者に入力を求める。

#### (F5)DOM構造比較用IEオブジェクト

オフライン中操作反映時のみ実行される。サーバ応答からバナー広告などをフィルタし(F5-1)、利用者操作ログ管理DBに保存されたオフライン中の応答とDOM構造を比較する。比較結果を(F4)オフライン中操作反映用IEオブジェクトに返す。

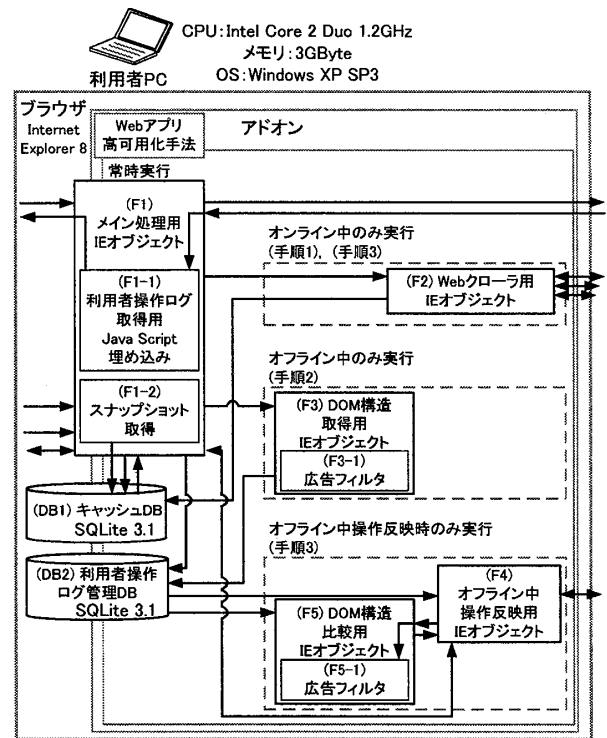


図2: 実装システムの概要

## 4. 実装システムによる提案手法[1]の評価

Google社が提供する、Ajaxを用いて実装された代表的なWebアプリの1つであるGmail[4]を対象に、3章に前述の実装システムを用いて、オフライン中の利用(受信メールの確認と新規メールの作成、送信)(手順2)と、オンライン後のオフライン中操作の反映(オフライン中に作成した新規メールの送信)(手順3)が可能であることを確認した。また、Gmailのログイン画面に接続する際、提案手法の導入により、従来の処理に追加される次の3つの処理に必要な時間(サーバとの通信時間を除く)の合計を測定した。(1) (F1-1)利用者操作ログ取得用JavaScript埋め込み、(2) クローラの起動、(3) キャッシュデータの作成。その結果、0.125秒であった。提案手法は、Webアプリの画面表示毎に、0.125秒の処理時間を追加することで、オフライン中のWebアプリの利用を可能とする。

## 5. おわりに

本稿では、ネットワーク接続が正常でない時にも、Webアプリケーションの利用を可能とすることで、その可用性を向上する手法を実装した。また、Ajaxを用いて実装された、代表的なWebアプリケーションの1つを対象に、実装システムにより、Webアプリケーションの高可用性が可能であることを確認した。さらに、Webアプリケーションの高可用性のため、提案手法は、画面表示毎に0.125秒の処理時間の追加が必要であることを実測により明らかにした。

[1] 荒井他, “ブラウザのキャッシュと利用者操作ログを用いたWebアプリケーション高可用性手法の提案,” 2010 情処全大。

[2] J.J. Garrett, Ajax: A New Approach to Web Applications, <http://www.adaptivepath.com/ideas/essays/archives/000385.php>

[3] SQLite, <http://www.sqlite.org/>

[4] Gmail, <http://mail.google.com/mail/help/intl/ja/about.html>