

Just in Time Proxy: 必要な時に必要なだけ使用できるプロキシ機能作成法

曾谷 俊男 青木 義則
日本アイ・ビー・エム株式会社 東京基礎研究所

近年、プロキシサーバ機能はセキュリティ確保のための代理アクセス機能を提供するだけでなく、翻訳や要約など様々な機能を提供するようになってきた。しかし、その機能を利用するためには、プロキシサーバの管理運営、Webブラウザの設定を必要とする。そのため、誰でも簡単にプロキシ機能を使い始めたり、切り替えたり、止めたりすることはできなかった。本論文では、プロキシ機能をウェブブラウザ上で動作させ、プロキシ機能の設定をウェブページ中から自動的に行う方式を述べる。本方式によって、ページの閲覧操作だけで、容易にプロキシ機能の提供を受けることができるようになる。

Just in Time Proxy: A Proxy Service for Immediat use

Toshio Souya and Yoshinori Aoki
IBM Research, Tokyo Research Laboratory

Recently, variety of functions are provided by proxy servers. Such serices are not only as a proxy which taking a part of a firewall, but filtering or embedding functions in pages. These proxy functions can be available by paying efforts for server maintenance and client program setting. It is hard to provide many proxy functions in a network because they require too much efforts. This paper describe a method for making proxy functions to be executed on web browsers, and to control it by a web page. End users can enjoy services by proxy functions created by this mehod.

1. はじめに

本論文は、Webアプリケーションにおいて、プロキシサーバ機能を提供する方法に関するものである。

プロキシサーバの機能は元々ファイアウォール機能を実現するためにクライアントか

らサーバへのアクセスを代行・中継するものであった[1]。しかし、近年では、次のような機能もプロキシサーバの機能として使用されるようになってきた。

- (1) リクエストの先取りによるプリフェッチ、オートパイロット、キャッシング
- (2) コンテンツのフィルタリング：内容の追加（プローブの埋め込み、ウイルス

チェック等) [2, 3]、変更(翻訳、要約、語調変換など)や形式の調整、削除(グラフィックの削除、色数調整、不適切なリンクの削除など)

このようなプロキシサーバの機能の多様化に伴い、従来のようにただ一つのプロキシサーバがネットワーク上にあれば充分というわけではなくなってきた。特にWebベースのアプリケーションが開発されるにつれ、特定のアプリケーションのための様々なプロキシサーバの機能を、素早く簡単に使い始め、同様に素早く簡単に使い終わる必要が出てきた。しかし、これに実現している例はない。

本論文で述べる方式はプロキシサーバ機能をWebアプリケーションの要求に基づいて「即座に」提供し上記の問題を解決するためのものである。つまり、Webアプリケーションが必要とした時に、通常のブラウジング以外の操作を必要とすることなくプロキシサーバ機能を使い始められる。必要なくなればプロキシサーバ機能を簡単に終了し、後はプロキシサーバ使用前と変わらずにネットワークにアクセスができる。

2. プロキシ機能構成法概要

本論文の方式は、次の2点から構成される。

- (1) プロキシサーバの実行環境としてWebブラウザを利用すること
- (2) Webページの読み込みと破棄動作によって、プロキシサーバ機能の制御を自動的に行うこと

本手法には次の効果がある。

従来、Webブラウザでプロキシの設定を行うには、単にWebページを閲覧する以上の知識が必要である。したがって、新たなプロキシサーバが必要な場合、その設定を一般ユーザーにやらせることはあまり現実的ではない。本方式では、Webページのダウンロードによって、プロキシサーバ機能の制御を自動的に行

うので、ユーザーはこのような面倒な設定から開放される。Webページのダウンロードは初ページ閲覧の初歩的操作なので、すべてのユーザーを対象とすることができる。クライアント側ではページ閲覧に伴ってプロキシサーバ機能の制御を自動的に行うので、プロキシサーバの動的な切り替えを簡単に実現できる。

またサーバマシン側では、複数のプロキシサーバを切り替えて使用する必要がある場合、使用する可能性のあるプロキシサーバをサーバマシン上ですべて常時稼働させておく必要があった。本方式では、必要な時に必要なだけ、クライアント上で動作させることが可能である。様々なプロキシが必要な場合でも、それらを常時サーバマシン上で稼働させておく必要がなくなり、それらの管理も必要なくなるという利点がある。サーバマシンの負荷を大幅に減らすことができる。

3. 実装方法

プロキシサーバの実行環境としてWebブラウザを利用するための実装方法としては、Javaアプレット、ActiveXコントロール、プラグインの3通りがある。Webブラウザ上で動作しながら、クライアントマシンのTCP/IPのソケット通信ができる実装方法であればよい。Javaアプレットとして実現する場合、セキュリティの制約を取り扱うための方策を講じて必要な通信ができるようにする。具体的には、署名付きアプレットとして実現する。

Webページのダウンロードによって、プロキシサーバ機能の制御を自動的に行うやり方はNetscape NavigatorではWebページ上でのイベントハンドラとスクリプト言語(JavaScript)を用いることができる。ダウンロード終了と同時に呼ばれるハンドラ関数によって、サーバ機能が立ち上がるようスクリプトを記述する。どの実装方法でも、Webページの読み込みと同時に変更し、破棄と同時

にプロキシの設定を元に戻すという機能は実現される。

4. 他の実装方式との比較

本方式で解決したプロキシ管理運営の手間の問題を従来技術で解決しようと試みるならば、次の2つの方法が考えられる。

方法1 サーバマシン上で必要なプロキシサーバ全てを動作させる。

方法2 クライアントマシン上で、ネイティブアプリケーションとして必要なプロキシサーバを動作させる。

サーバ上にすでにプロセスが立ち上がっている場合は、Webブラウザ上のプロキシ設定さえできるのであれば、即座にプロキシサーバ機能が使用可能と言える。しかし手動でのプロキシサーバ切り替え作業を多くのユーザにやらせることは現実的ではない。サーバ上ですでにプロセスが立ち上がっていない場合は、権限の問題からプロキシサーバを勝手に立ち上げることができないのが普通であり現実性に欠く。

一方クライアント上でプロキシサーバ機能を稼働させようとする、クライアントマシンにインストール作業をしなくてはならない。使用、不使用によって手動でブラウザの設定を変更する作業が必要になる。これらの作業はエンドユーザが行う他なく、現実性に欠く。

5. 実装および応用例

プロトタイプの実装例を次に示す。但し、セキュリティ関連のJavaAPIが異なることと、JavaScriptの互換性の問題から、このプロトタイプはNetscapeのみで動作する。

(1) プロキシサーバ機能は署名付きアプレットの形で実装した。署名付きにしたことで、セキュリティの制限を越えることができる。その他の実装は、一般的なマルチスレッドのプロキシサーバの実装であ

る。

(2) 上で実装したプロキシサーバ機能を制御する機能を、JavaScriptで実現しHTMLページを作成した。プロキシサーバ機能を同時にダウンロードするために、同ページにプロキシサーバアプレットを埋め込んだ。ページロードとアンロード時の具体的な動作は次の通り。

i. ページロード完了時

OnLoadイベントハンドラで呼び出されるJavaScript関数に次の機能を実装する。

- ・ブラウザの各種設定が読めるよう、セキュリティ設定をする
- ・ブラウザのプロキシ設定を読み取る
- ・設定を保存する。
- ・同じ設定をプロキシサーバのアプレットに渡し、これを起動する
- ・ブラウザがアプレットを参照するように、ブラウザのプロキシ設定を変える

ii. ページアンロード (閉じる) 時

OnUnloadイベントハンドラで呼び出されるJavaScript関数に次の機能を実装する。

- ・上で保存しておいたプロキシ設定通りにブラウザのプロキシ設定を戻す

1. 上で作成したプロキシサーバ機能とそれを制御するWebページをWWWサーバに登録し、参照可能としておく (図 1 a)。このサーバは必要なクライアントからHTTP経由でアクセス可能であればよく、ファイアウォールを介していても問題はない。

以上で、プロキシサーバ機能を利用するための準備が完了する。次にダウンロード、実行される場面での動きを述べる

1. プロキシ機能が必要になると、ユーザの

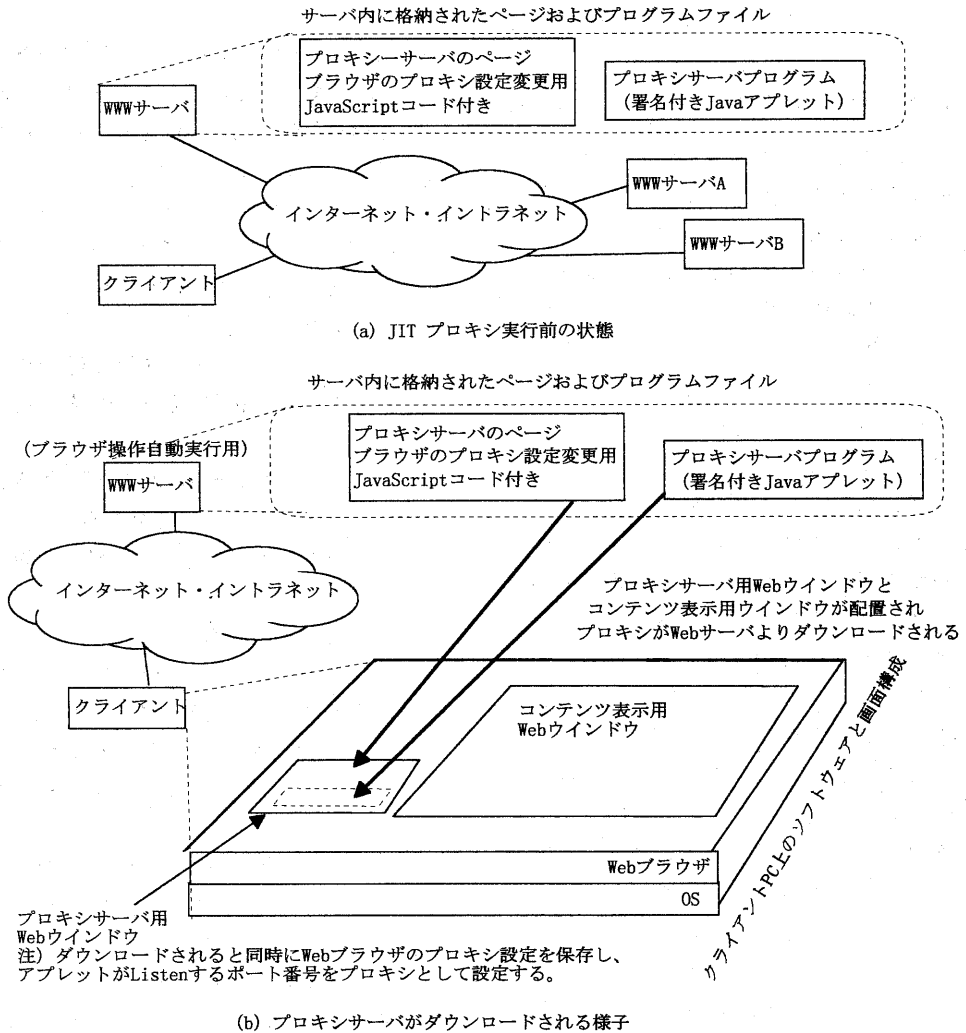
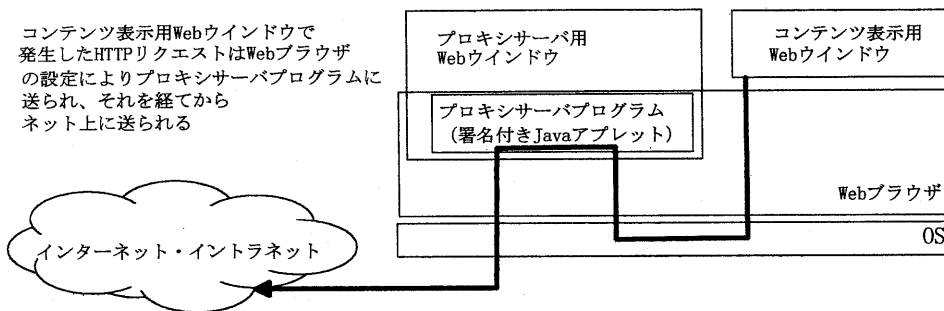
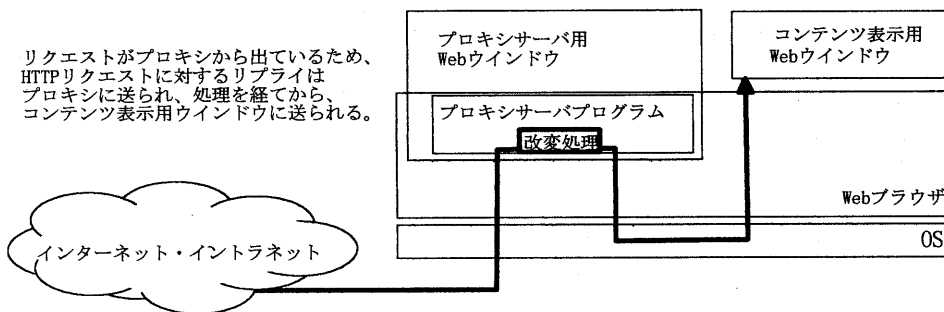


図1 JITプロキシ実行前と、ダウンロード後に実行中の様子

- リンククリックやプロキシサーバと連携して動作するWebアプリケーションからプロキシページ用のWebブラウザウインドウが作られ、そこにアプレットと共にプロキシページがロードされる(図1b)。
2. プロキシページがロードされると、プロキシ中のJavaScriptプログラムがブラウザのプロキシ設定をプロキシページ中から変更してプロキシページ中のプロキシ機能を参照するようにする。これ以降、
 - コンテンツ表示用のWebブラウザウインドウからのHTTPリクエストはすべてプロキシページ中のプロキシ機能を経由することになる(図2a)。返されるHTTPリプライも全て同プロキシを経由し、ここで内容のフィルタリングや、置き換えが行われる(図2b)。
 3. これらのプロキシ機能が不要になれば、アプリケーションから自動的にプロキシページを閉じるか、ユーザがウインドウ



(a) HTTPリクエストの流れ



(b) HTTPリプライの流れ

図2 HTTPリクエストとリプライの流れ

クローズの操作をやってもらう。その時にブラウザのプロキシ設定は、プロキシページの機能により復元され、プロキシ機能は動作を終了する。

6. おわりに

本論文では、必要な時に必要なだけ利用可能なプロキシ機能の構成法について述べた。このようなプロキシ機能は、プロキシプログラムの動作環境としてWebブラウザを使用すること、プログラムの設定をWebページから行うこと、の2点から実現される。本構成法により、より使いやすいサービスが提供できるようになると考える。今後は、この特性を生かしたサービスの実装と実証を行っていききたい。

参考文献

- [1] "ファイアウォール構築 インターネットセキュリティ", D. Drient Chapman, Elizabeth D. Zwicky, 歌代 和正 監訳、オライリージャパン
- [2] Aoki, Y., Ando, F., Nakajima, A., "Web Operation Recorder and Player," IBM TRL Research Report RT0267, 1998.
- [3] Aoki, Y., Nakajima, A., "User-Side Web Page Customization," Proceedings of 8th International Conference on Human Computer Interaction (HCI International '99), Vol. 1, pp. 580-584, August 1999.