Cloud for Client 技術：無限タブ Web ブラウズへの挑戦

新田 竜規 † 新井 イスマイル ††　榎堀 優 † 西尾 信彦 †
† 立命館大学情報理工学部　†† 立命館大学総合理工学研究機構

1 はじめに

近年，GoogleDocs[*1]や Splashup[*2]など，スタンドアローンアプリケーションと比べて操作性において遜色のない Ajax を用いている等の動的な Web アプリケーションが出現している．動的な Web アプリケーションとは，Web サーバ上の CGI と Web ブラウザ上の JavaScript が協調動作するアプリケーションのことであり，ローカル環境へのインストールを必要とせず，Web ブラウザのみを用いてアプリケーションを利用することができる．たとえば，メールクライアントと同様の機能を提供する GMail[*3]のような Web メールなどが典型的な例と言える．つまり，ユーザは Web ブラウザの一つのドキュメントを，一つのアプリケーションのような感覚で利用できるようになってきている．

こうした中，Web アプリケーションを複数同時に利用したい要求がある．また Web アプリケーションに限らず，Web ブラウジングを行っている際は閲覧や作業の途中にあるドキュメントその状態を残しておきたい要求がある．しかし，ドキュメントをたくさん生成するとその量に比例してメインメモリを消費してしまうため，PC に負荷がかかってしまい，PC の性能限界を超えてドキュメントをたくさん生成し快適に利用することはできない．

一方，買い換え後の古い PC は活用されることがなく，これを余剰リソースと捉えると，現在利用している PC のリソースとして活用したい要求がある．

本研究では，余剰 PC を活用することによって，ユーザが利用する PC の性能限界を超えたドキュメント数による Web ブラウジングを実現することを目的とする．また、本研究で目指す Cloud for Client 技術とは，通常サーバサイドで利用される Cloud 技術をクライアントが直接に恩恵を獲得できるようにするためのもので，サポートするホストを用意すれば，その分 scalable に直接にクライアントホストの能力を向上できるものである．本稿では，その中から無限にプロセスを生成できる機構を Web ブラウザのドキュメント数を無限に増やすことに置きかえて，その実現機構について述べる．なお，現在 Web ブラウザのほとんどが，“タブ” を用いて複数のドキュメントを 1 つのウィンドウ内で管理するタブブラウザであることから，本稿では主にタブをドキュメントとして扱う．

†　TATSUNORI NITTA
††　ISMAIL ARAI
†　YU ENOKIBORI
†　NOBUHIKO NISHIO
College of Information Science and Engineering, Ritsumeikan University (†)
The Research Organization of Science and Engineering, Ritsumeikan University (††)
*1 http://docs.google.com/
*2 http://www.splashup.com/
*3 http://mail.google.com/

2 既存研究

PC への負荷を軽減する方法として，プロセスマイグレーションが研究されている [1][2] ．プロセスマイグレーションとは，プログラムを停止させずにその動作状態を保ったまま他の PC へ移動させ，そこで実行を継続させることである．利用に当たっては，プロセスマイグレーションに対応した OS への更新，ミドルウェアのインストールが必要で，他に利用しているスタンドアローンアプリケーションの互換性がなくなる恐れがある．

本研究では，クライアントノード（ユーザが利用する PC）に対しては Web ブラウザのみに手を加え，タブをリソースノード（クライアントノードとは別の PC）上に複製し，非閲覧中のタブのリソースを解放することで，クライアントノードの性能限界を超えてタブを生成し快適に利用することを可能にする手法を提案する．

3 提案システム

本システムの構成を図 1 に示す．本システムでは，以下の 3 つの要素から構成される．

- クライアントノード
  ユーザが利用する PC．Web ブラウジングを行う．
- リソースノード
  ユーザが利用していない余剰 PC．クライアントノード上の Web ブラウザで生成されたタブの複製先．
- 管理サーバ
  本システムにおいて，各ノードを自由に追加・削除できるようにするため，ノード情報・タブ情報の管理を行う．また，クライアントノード・リソースノードに対するプロキシの役割も果たす．

ユーザがクライアントノードで Web ブラウジングを行う．このときクライアントノードの Web ブラウザ上でユーザが行った操作情報（キー入力・マウス操作といった入力，新規タブ作成・タブ選択といったブラウザイベント）をリソースノードに送信し，その操作をリソースノードの Web ブラウザが行うことで，クライアントノードと同様の状態のタブをリソースノードに複製することができる．また，

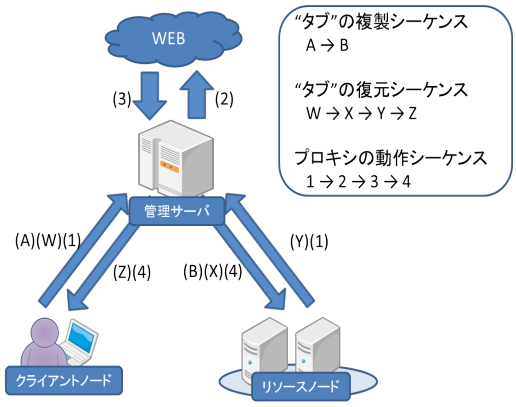

図 1: システム構成

ユーザが閲覧中以外のタブのリソースをクライアントノードより解放することで，クライアントノードへの負荷は軽減される．さらに，一度リソースを解放したタブを再度閲覧する際は，リソースノードのタブのストリーミング映像を再生する．このとき，VNC などのデスクトップ共有システムのようにクライアントノード上の操作をリソースノードに反映し，クライアントノードからリソースノードを遠隔操作することで，タブの復元を可能にする．

以下に各動作の詳細を説明する．

- タブの複製

  クライアントノードは，自身が実行するブラウザのタブ上で行われたキー入力・マウス操作といった入力，新規タブ作成・タブ選択といったブラウザイベントを，ノード情報を管理する管理サーバに送信する（図 1（A））．管理サーバはクライアントノードより受信した情報を，リソースノードに送信する（図 1（B））．リソースノードは受信した情報を，自身が実行するブラウザに反映することでクライアントノードと同様の状態のタブをリソースノード上に複製することができる．

- タブの復元

  クライアントノードから管理サーバにタブの復元リクエストがあった場合（図 1（W）），そのタブの複製を保持しているリソースノードに復元リクエストを送信する（図 1（X））．復元リクエストを受信したリソースノードは，該当のタブのストリーミング映像を管理サーバを経由して（図 1（Y））クライアントノードに送信する（図 1（Z））．このとき，クライアントノードで行われた操作をリソースノード上の該当タブに反映する．その反映後のストリーミング映像をクライアントノードに送信し続け，クライアントノードからリソースノードを遠隔操作することで，擬似的にタブの復元を可能にする．また，URL の更新が行われる際にストリーミング映像の送信を停止し，タブの複製時の動作に戻ることで，完全にクライアント上で動作するプロセスに復元できる．

- ノードの管理

  本システムでは，クライアントノードとリソースノードの二種類のノードが存在する．これらのノードは管理サーバで管理され，自由にノードの追加や削除ができる．クライアントノードからリソースノードへタブの複製が行われるとき，管理サーバは残りリソースに余裕があるリソースノードに複製タブを生成する．

- プロキシ

  クライアントノードとリソースノードが同様の動作を行うことで，URL の更新を行う際にクライアントノードとリソースノードから同一の Web サイトへの重複リクエストが送られることになる．そのリクエストが掲示板への投稿など Web サイトに変更を及ぼすものの場合，二重書き込みが発生してしまう．
  プロキシはこの重複リクエストを認識して（図 1（1）），Web サイトに対して一度だけリクエストを送信する仲介を行う（図 1（2））．リクエストに対する Web サーバからのレスポンスを受け取った管理サーバは（図 1（3））， リクエストがあったクライアントノードとリソースノードそれぞれにそのレスポンスを送信する（図 1（4））．

4　実装状況と今後の課題

現在，クライアントノード・リソースノード上で動作させる Web ブラウザ機能を GoogleChrome のエクステンションとして実装中で，また管理サーバの機能についても一部実装が完了し，ノードの管理機能が実現できている．今後は，クライアントノードで行った入力をリソースノードに反映させる機能，タブの復元機能，プロキシ機能を実装する．

参考文献

[1] D.S. Milojicic, F. Douglis, Y. Paindaveine, R. Wheeler, and S. Zhou. Process migration. *ACM Computing Surveys*, Vol. 32, No. 3, pp. 241–299, 2000.

[2] YOSUKE YOKOYAMA, YOSHIHIRO OYAMA, and AKINORI YONEZAWA. Os extension for transparent migration of server processes. *IPSJ SIG Notes*, Vol. 2005, No. 48, pp. 77–83, 2005-05-25.