

# リッチクライアント技術を用いた仮想マシンモニタの提案と実装

高橋 一志†

山本 知仁†

## 1. はじめに

近年、計算機の小型化、通信インフラの整備が進むにつれ、遠隔地にある計算機上のデスクトップ環境を、手元のコンピュータ上に再現するような技術が普及しつつある。このようにして再現されたデスクトップは一般に仮想デスクトップと呼ばれ、これを提供するためのインフラストラクチャをVDI

(Virtual Desktop Infrastructure) と呼ぶ。VDI は個人用計算機の新たな集中管理アプローチとして注目を浴びている。有名なVDI としては、Sun Microsystems のSun Ray[1], Microsoft のWindows Terminal といったシンクライアントが挙げられる。これらの技術は、既存OSのデスクトップをそのまま仮想化しているため、アプリケーション側に特別な処理は必要ない。しかしながら、これらのシンクライアントは専用端末と、専用プロトコルを使用しており、基本的にクローズドなネットワークでの運用を前提としている。

一方で現在、シンクライアントとは別に汎用的なコンピュータと一般的なプロトコルを利用してVDI を実現するシステムが登場しつつある。WebShaka によって開発されたWeb アプリケーションであるYouOS[2]は、Ajax (Asynchronous JavaScript+XML : Ajax) を使用して、Web ブラウザ上で一見しただけでは、既存OS 上のデスクトップ環境と峻別することのできないユーザビリティを備えたデスクトップを提供している。そのためYouOS はVDI の一種であると見ることができる。われわれはこのような仮想的なデスクトップをWeb ブラウザ上で動作させることのできる技術を新たにWebVDI と呼ぶことにする。

WebVDIは基本的にHTTPD を介して動作するサーバサイド(と、場合によってはWeb ブラウザ上で動

作するJavaScript) のプログラムである。既存のデスクトップアプリケーションとは構築方法も、動作プラットフォームも異なる。そのため既存のソフトウェアをWeb アプリケーションとして再利用することは基本的に不可能であり、再利用のためにはソフトウェアの再実装の作業が必要となる。そこで本論文では仮想マシンモニタ (Virtual Machine Monitor : VMM) とサーバサイドアプリケーションを組み合わせることにより、既存のOS をWebブラウザ上で動作させ、その上でソフトウェアを動作させるという新たなWebVDI の方式を提案する。

## 2. リッチクライアントを用いた VMM

Fig1にシステム概念図を示す。本研究ではVMM とサーバサイドアプリケーション、Web ブラウザ上のJavaScript を組み合わせ、Web ブラウザ上でOS を動作させることにより、OS が持つデスクトップを仮想化する。本システムはユーザが操作することになるクライアントを担うWeb ブラウザ、そして一般的なHTTPD と、その上で動作する簡単なCGI スクリプト、さらにこのシステムのコアとなるVM を制御するためのVMM から成る。クライアントには特殊な機能拡張などは必要なく、一般的なHTML と JavaScript でかかれたコンテンツをいつでも Server からダウンロードできる。ブラウザはマウスの座標と、キーボードのイベントを逐一把握しており、ブラウザ上にてイベントが発生するとAjax による非同期通信にてHTTPD 経由でCGI スクリプトに伝えられる。CGI スクリプトは定期的にそれらのイベントをVMMに伝え、VMMはそのイベントをVMへと伝える。VMはイベントに応じた処理を行い、その結果はVGA Framebuffer に反映される。出力されたVGA Framebuffer を必要に応じてブラウザへ返すことでブラウザ上にVM のデスクトップを再現すること

---

† 金沢工業大学 工学部 情報工学科

ができる。これによりWeb ブラウザ上であたかもソフトウェアが動いているかのように振る舞わせることができ、また、VM 上にWeb アプリケーションとして振る舞わせたいソフトウェアをインストールしておけば、ただちにそれをWeb ブラウザ上で操作することが可能となる。

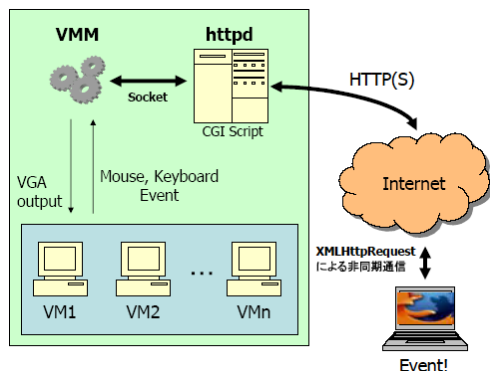


Fig.1 システム概念図

### 3. システムの構成要素

今回のシステムを構築するにあたって、それぞれのコンポーネントを試験的に実装した。コンポーネントの詳細は以下のとおりである。

**VMM** 本研究で開発する VMM はいわゆる Hosted VMM である。本 VMM は Intel VT-x [3] を使用した Full-virtualization を実現しているため、VMM により生成される VM の ISA は実マシンの IA-32 とほぼ同等である。そのためゲスト OS に何ら変更点を加える必要がない。現在の実装ではホスト OS に Linux Kernel 2.6.18 をサポートしている。実装は C 言語で行われており約 10,000 行ほどである。

**Managed Server** このプログラムは、RFB Protocol [4] を使用して直接 VM と通信を行うためのものである。1 台の VM ごとにひとつの pthread を生成し、pthread 内で VM と、それを利用している端末 (Web ブラウザ) を管理する。VM の VGA Framebuffer や、ユーザのマウス、キーボードイベントはすべてここに集めら

れ、ここから改めてユーザや VM へと伝えられる。実装は C++ で行われており、約 4,000 行ほどで実現されている。

**Fabricator** Fabricator は CGI Script であり、HTTPD を通じて、Web ブラウザ上の JavaScript と通信を行うためのものである。このプログラムは Managed Server からのデータを、Web ブラウザや JavaScript が解釈できる形にフォーマットするためのフォーマッタである。また Managed Server とのデータやりとりは一般的な TCP ソケットを使用して、通信プロトコルには独自プロトコルを用いる。実装は Common Lisp で行われており、400 行ほどである。

**Web-browser Interface** Web ブラウザへの VGA Framebuffer への表示、キーボードやマウスのイベントの取得などを行う。これらのデータは Ajax を用いて、非同期に CGI Script へと伝えられる。JavaScript 部分は 400 行未満と非常に小さく、多くの Web ブラウザでも互換性が高くなるように設計されている。

### 4. おわりに

本研究では、Web ブラウザ上で高度なデスクトップ環境を提供する Web アプリケーションを、新たに WebVDI と定義し、さらに VMM と Web ブラウザを用いて、Web ブラウザベースのシンクライアントという新たな WebVDI の形を定義した。今後の課題としては、システムの有用性を示すための定量的、定性的な評価である。

#### 参考文献

- [1] B. K. Schmidt, M. S. Lam, and J. D. Northcutt .: The Interactive Performance of SLIM: A Stateless, ThinClient Architecture, Proceedings of the 17th ACM Symposium on Operating Systems Principles, Kiawah Island Resort, SC, Dec. 1999.
- [2] WebShaka, Inc .: <https://www.youos.com/>
- [3] Intel Corporation .: Intel 64 and IA-32 Architectures Software Developer's Manual Volume3B System Programming Guide, Part 2.
- [4] Olivetti Research Ltd / AT&T Labs Cambridge .: The RFB Protocol.