

B-034

仮想計算機の画面転送処理効率化のための Windows ディスプレイドライバ Windows Display Driver for Efficient Screen Transfer on Virtual Machine System

坂井 成道[†]
Narumichi Sakai

後藤 真孝[†]
Masataka Goto

峰松 美佳[†]
Mika Minematsu

西林 泰如[†]
Yasuyuki Nishibayashi

村井 信哉[†]
Shinya Murai

1. はじめに

我々は、ネットワークの常時接続を前提に、映像や音声などの入出力機能とデータの通信機能に特化したディスプレイ端末 (ND 端末) からネットワーク上のサービスを利用する、ネットワークディスプレイシステム (ND システム) の開発を行っている。サービスの一つに PC アプリケーションを想定しており、従来の個人 PC への接続に加え、仮想計算機 (VM) への接続も実現することで、PC サービス環境のサーバ化を実現した[1]。また、画面伝送の効率化手法であるページタイトルフレームバッファ (ページタイトル FB) を提案し、Linux ゲスト OS における試作を行った[2]。

本稿では、Windows ゲスト OS 向けのページタイトル FB の試作とその評価について述べる。第2章では ND システムと PC 画面転送の概要、ならびにページタイトル FB について述べる。第3章では Windows ディスプレイドライバアーキテクチャと Windows ゲスト OS におけるページタイトル FB の試作について述べ、第4章で評価を行い、最後にまとめる。

2. ND システムの PC 画面転送とページタイトル FB

ND 端末を用いることによりユーザはネットワーク上のサーバに接続して、各自のデスクトップを利用することができる。接続先のサーバ上では VM が稼動しており、1つの物理マシン上で複数の VM を動作させることで複数のユーザが PC 環境を利用可能である。図1に仮想計算機を用いた ND システムの装置構成を示す。

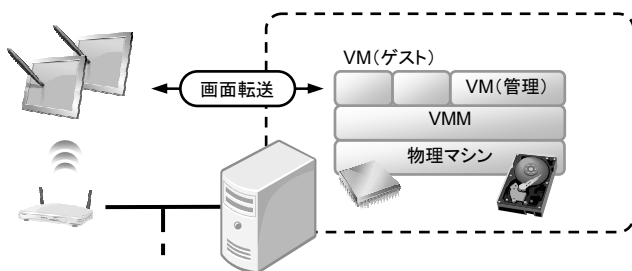


図1 仮想計算機を用いた ND システム装置構成

ユーザが ND 端末を操作し、その結果として画面に更新があると、VM は自身が保持しているフレームバッファ (FB) の値を変更する。この変更を VM モニタ (VMM) とデバイスエミュレータが検出し、該当部分の画像を圧縮し ND 端末へと転送する。

画面の更新部分の検出方法として、更新前後の FB (画面のピクセル) の値を比較する方法がある。しかし、こ

[†] (株) 東芝 研究開発センター R&D Center, Toshiba Corporation

の従来方法では、画面サイズと更新頻度に比例して処理負荷が増大するため、我々は FB のアドレス配置を変更してメモリ管理ユニット (MMU) の更新ビットを利用する方式、ページタイトル FB を考案した。

MMU の更新ビットを利用すると、FB に該当するメモリ領域の更新がないページの値の比較を省くことが可能である。しかし、画面の横方向1ラインが1つのページに対応している通常の FB の構造 (リニア FB) では、横幅が画面と比較して小さい矩形の画面更新の場合、ページ内の水平方向の値の比較に無駄が生じる。

そこで我々は、32bpp の縦横 32 ピクセルの正方形をタイトルとし、これを1ページに対応させることで比較対象となるページ数を削減した。さらに、タイトルを縦に並べて短冊を作り、それを横に敷き詰める形にすることで処理の複雑化を抑えている。

図2に、2次元平面である画面のピクセルの並びから1次元のメモリ領域へのマッピングを濃淡で示す。色が薄いほど連続したメモリ空間上の後方に位置していることを意味する。アイコンが描画されると、波線部のページの更新ビットが1になる。このように、矩形の画像が描画された場合、リニア FB よりもタイトル FB の方が、比較対象であるページ数が少ない。

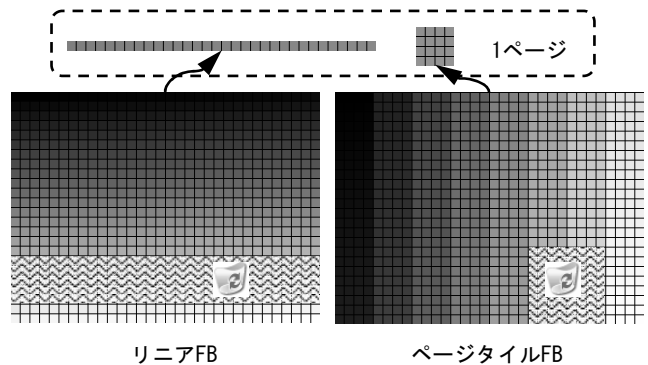


図2 リニア FB とページタイトル FB

3. 実装

ページタイトル FB の実装には、ゲスト OS の描画プログラムとデバイスエミュレータの両方を改変する必要がある。Windows ゲスト OS に対応するために、Windows の画面描画をつかさどるコンポーネントの1つである DDI (Display Driver Interface) ドライバを改変した。

ほとんどの場合、Windows のアプリケーションソフトウェアは Win32 サブシステムが提供する関数の組み合わせとして構築されている。ユーザからの要求は Win32 関数を通じてシステムサービスに伝えられ、それが描画操作であった場合は GDI (Graphic Driver Interface) の関数が描画を行う。DDI ドライバは GDI の補佐の役割を果たす DLL (Dynamic Link Library) で、ディスプレイドライバ開発

者が必要に応じて用意することができる。DDI ドライバは、GDI からの呼び出しに応じて、ミニポートドライバが提供する FB に描画を行う。図 3 に Windows のディスプレイドライバアーキテクチャを示す。

今回の実装では、DDI ドライバの描画関数の一つであり、Windows のグラフィカルユーザインターフェースのほとんどの描画で利用されている DrvCopyBits 関数を改変することのみでページタイル FB を実現した。このため、スクロール等の画面から画面へのコピー操作など、一部の描画に対しては未対応である。実装のベースに用いた DDI ドライバは Windows Device Driver Kit (DDK) [3] に付属の framebuf.dll を利用した。DDI ドライバには対となるミニポートドライバが必要であるが、今回は Windows とドライバ互換性のある OS である ReactOS[4] の vbemp.sys を利用した。

VMM は Xen3.1[5] を使用した。ページ単位の比較が行えるように vga.acc.patch を当て、ioemu を改変してタイル FB を扱えるようにした。

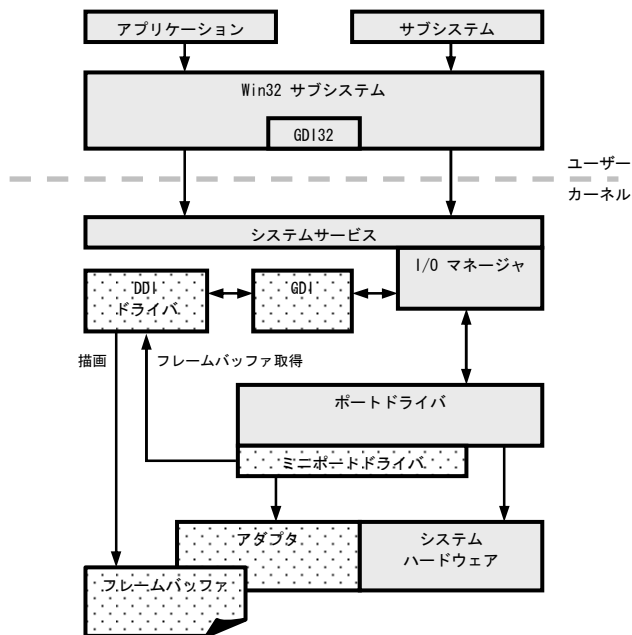


図 3 Windows ディスプレイドライバアーキテクチャ

4. 評価

リニア FB、および、ページタイル FB のそれぞれについて Flash 動画再生を行い、フレームレートと処理時間について計測を行った。処理時間には、VMM が FB の更新を検出し、更新部分を圧縮し、圧縮したデータをネットワーク送出のバッファにコピーするまでの時間を計測した。VM を動作させたサーバは、2CPU(Xeon 5110 1.6GHz FBS 1066MHz)、メインメモリ 4GB である。VM 上で動くゲスト OS として Windows XP を用いた。動画コンテンツのオリジナルフレームレートは 30fps、サイズ 480 ピクセル×360 ピクセル、再生時間 2 分 20 秒程度である。

リニア FB・ページタイル FB を用いた場合の 1 フレームあたりの処理時間とフレームレートを比較した。図 4 は、Windows でページタイル FB とリニア FB を採用する方式

を用いたときの処理時間の比較を示しており、リニア FB よりも処理時間が短縮されていることがわかる。図 5 の Linux の場合と比較すると、同程度短縮されているといえる。図 6 に示すように、処理時間の短縮は、フレームレートの向上にも反映されることが確認できる。なお、フレームレートの向上は処理時間の短縮と単純に比例関係にはならない。これは、計測した処理時間以外にプロトコル処理やタイマー制御の時間が含まれることが原因である。

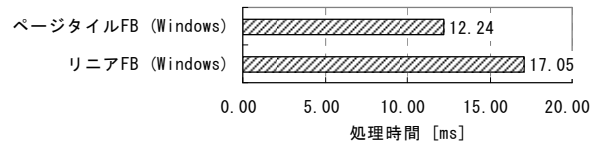


図 4 処理時間比較 (Windows)

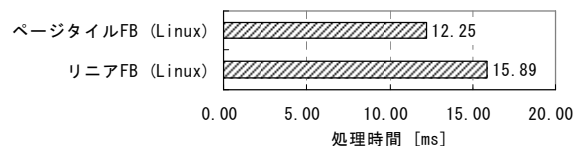


図 5 処理時間比較 (Linux)

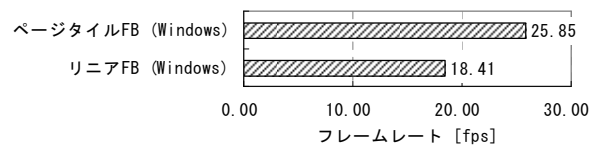


図 6 フレームレート比較

5. おわりに

本稿では、仮想計算機サーバの画面転送処理効率化手法であるページタイル FB の、Windows ゲスト OS への適用方法と評価について述べた。リニア FB とページタイル FB の Windows の画面描画性能の評価を行い、ページタイル FB を利用した場合、リニア FB と比較して画面更新領域の検出にかかる負荷が軽減され、Flash 動画コンテンツ再生のフレームレートが向上することがわかった。ページタイル FB を用いることにより Linux と Windows のどちらにおいても画面転送の性能が向上することを示した。

参考文献

- [1] 峰松 美佳, 後藤 真孝, 西林 泰如, 村井 信哉, “ネットワークディスプレイシステムにおける仮想計算機サーバの試作”, 2008 年信学総大, D-6-14, Mar. 2008.
- [2] 後藤 真孝, 西林 泰如, 峰松 美佳, 村井 信哉, “仮想計算機環境における画面伝送処理の効率化”, 2008 年信学総大, D-6-13, Mar. 2008.
- [3] “Windows Device Driver Kit”, <http://www.microsoft.com/japan/whdc/DevTools/ddk/default.mspix>
- [4] “ReactOS”, <http://www.reactos.org/>
- [5] “XenSource”, <http://xen.org/>