

ハイパバイザによる広告表示 Hypervisor-based Advertising

大山 恵弘[†]

Yoshihiro Oyama

1 はじめに

情報通信機器の画面が持つ広告表示先としての価値は高まる一方である。広告表示のための様々な方法が既に存在する。動画サービスの YouTube では、ウェブブラウザ内に表示する動画に広告を重ね合わせる。Apple による iAd は、様々なアプリケーションの画面内に広告を表示することを可能にする。また、Web サイトの画面内やポップアップにバナー広告を入れる伝統的な方法も広く用いられている。

これらの方法には、広告主にとって、いくつかの問題がある。第一は、広告を表示する場所が、動画内やブラウザのウィンドウ内に限られる点である。他のアプリケーションの利用時には広告は表示されない。また、表示されるとしても、ウィンドウ内などの狭い場所に表示せざるを得ないことが多い。第二は、広告を表示させるための仕組みが、OS、ブラウザ、ブラウザアドオンなどに依存することが多い点である。第三は、広告表示を制限するための設定やソフトウェアにより、ユーザが広告表示を抑制できる点である。

そこで、本研究では、OS やアプリケーションにほとんど依存せず、画面上の任意の場所に任意のサイズの広告を表示させるシステム ADUG (ADvertising from UnderGround) を提案する。OS のユーザは、たとえ OS 管理者の権限を有していても、ADUG による広告表示を抑制することができない。ADUG はハイパバイザとして実現されている。ハイパバイザとは、仮想マシンを提供し、その上で OS (ゲスト OS) を動作させることを可能にするソフトウェアである。著名なハイパバイザとしては Xen や VMware ESX がある。ADUG はゲスト OS とは独立に表示処理を実行するので、OS のユーザが利用中のアプリケーションに関係なく、確実に画面に広告が表示される。

2 提案システム

ADUG はゲスト OS の下で動作し、仮想マシンを管理している。ADUG は、ゲスト OS および仮想マシンの状態がある条件を満たしたタイミング (後述) で、画

面上に広告の画像や文字を表示させる。ADUG によって広告を表示した画面を図 1 に示す。この例では、電気通信大学を宣伝する「UEC TOKYO」というロゴを表示している。

ADUG の管理者と、ゲスト OS のユーザや管理者は異なることを仮定している。たとえば、組織のある部署のシステム担当者がその部署の全メンバーの PC に ADUG とゲスト OS をインストールする。ADUG (ハイパバイザ) の管理者権限はシステム担当者が保持し、ゲスト OS の管理者権限は各メンバーに渡す。また、インターネットカフェで提供される PC において、ADUG とゲスト OS はカフェの店員がインストールし、客には一般ユーザ権限だけを渡す。これらのケースにおいて、部署やカフェは、本システムの導入にあたり広告主や広告代理店と契約を結び、インストール数や稼働時間などに応じた収入を得る。

ADUG は BitVisor [1] のバージョン 1.0 を改造して実装されている。BitVisor は準パススルー型の構成をとるハイパバイザである (図 2)。BitVisor は OS 上ではなくハードウェア上で直接動作する。BitVisor は、ゲスト OS を制御しながら実行するが、制御は最小限に抑えられている。BitVisor は、セキュリティなどのために捕捉、制御の必要がある処理のみを、捕捉、制御する。ゲスト OS とハードウェアとの間のやり取りの大部分は、捕捉されたり、制御されたりすることがない。特に BitVisor はゲスト OS による画面制御関連の操作を捕捉しない。すなわち、ゲスト OS は、自身が持つデバイスドライバによってグラフィックハードウェアを制御する。それは ADUG においても同様である。

広告を表示するタイミングとしては、様々なものが考えられる。たとえば、10 分経過ごとに 5 秒間だけ表示させるなどの時間ベースの方法が考えられる。また、ハードディスクに書き込まれるディスクブロックに、事前に指定した文字列が含まれる場合に、その文字列に関連した広告を表示させるという文字列マッチングベースの方法も考えられる。最も実装が単純なものとしては、ハイパバイザが VMEXIT イベントを一定回数処理するたびに広告を表示するという方法もある。VMEXIT は仮想マシンからハイパバイザに制御が

[†]電気通信大学, The University of Electro-Communications

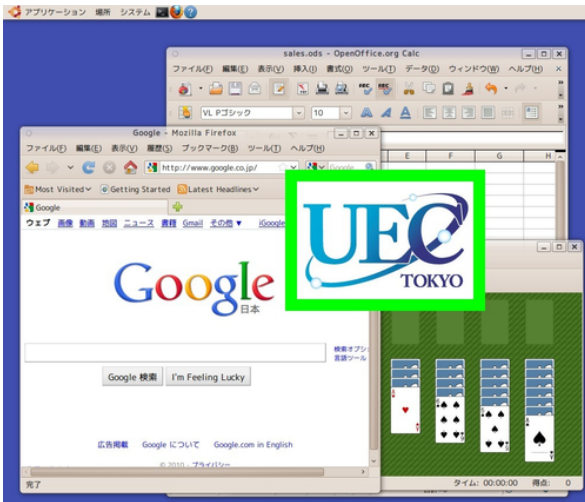


図 1: ADUG により広告を表示した画面

移るイベントである。この方法では、仮想マシンがある程度の量の処理をするたびに広告が表示される。

画面情報の更新は、幾世らによる研究 [2] の方法によって実現している。すなわち、グラフィックハードウェアが持つフレームバッファを ADUG が直接書き換えることによって実現する。具体的には以下の流れで画面情報を書き換える。まず、PCI ハードウェアから設定情報を読み取り、グラフィックハードウェアの制御レジスタ領域およびフレームバッファ領域のアドレスを得る。次に、その制御レジスタ領域への読み書きを通じて、VRAM 領域のアドレスを得る。その後、VRAM 領域のメモリを直接書き換えることにより、画面に画像や文字を表示する。広告として表示させる画像データとフォントデータは、ADUG のソースコードに埋め込む。これらの処理はハードウェアにのみ依存し、ゲスト OS に依存しないため、多様なゲスト OS 上に広告を表示できる。

ADUG による画面情報の更新は、ゲスト OS による画面情報の更新を全く考慮せずに行われる。よって、ゲスト OS と ADUG とで更新処理が競合する可能性はある。ただし、競合した場合でも、画面表示が崩れる程度の影響があるだけであり、システムがクラッシュするなどの深刻な障害は発生しないので、致命的な問題にはならないと考えている。

ADUG は自らの広告を消す処理は実行しない。OS のユーザは広告の上にウィンドウなどを新たに表示することにより広告を消すことができる。ADUG の拡張により、ユーザに広告を消させないようにすることは可能である。最も単純にそれを実現するには、非常に短い時間間隔で繰り返し広告を表示すればよい。

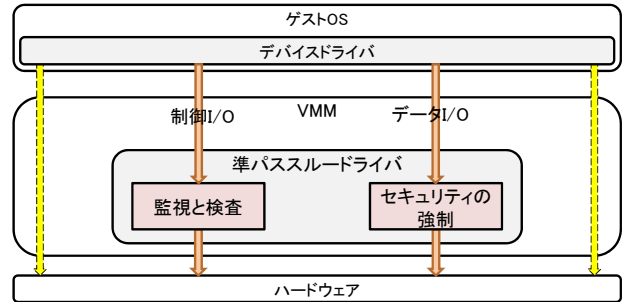


図 2: BitVisor における準パススルー型アーキテクチャ

現在実装に用いているハードウェアは、CPU が Intel Pentium D、チップセットが Intel 945G である。現在実験に用いているゲスト OS は Ubuntu 9.10 Linux である。実装コードは上記チップセットに依存している。

3 現状と今後の課題

現在、ADUG のプロトタイプを実装した段階である。そのプロトタイプは、一定回数 VMEXIT イベントを処理するたびに決まった画像を表示させる機能だけを有する。我々はそのプロトタイプを用いた実験を行い、ハイパバイザにより画面に広告を表示させることが可能であることを確認した。

今後の課題は、広告を表示させるタイミングを洗練させることである。たとえば、ゲスト OS のユーザが重要な仕事をしていないと推測されるときに広告を表示する技術や、画面上の重要でないと推測される領域に広告を表示する技術を開発する予定である。

謝辞 本研究の遂行にあたり、奈良先端科学技術大学院大学の幾世知範氏と豊田工業高等専門学校の平野学氏より有益な技術情報と助言をご提供いただいた。ここに深く感謝の意を表す。

参考文献

- [1] Shinagawa, T., Eiraku, H., Tanimoto, K., Omote, K., Hasegawa, S., Horie, T., Hirano, M., Kourai, K., Oyama, Y., Kawai, E., Kono, K., Chiba, S., Shinjo, Y. and Kato, K.: BitVisor: A Thin Hypervisor for Enforcing I/O Device Security, *Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE 2009)*, pp. 121–130 (2009).
- [2] 幾世, 平野, 品川, 奥田, 河合, 加藤, 山口.: 仮想マシンモニタ BitVisor のためのロールベースアクセス制御機構の設計と実装, 情報処理学会研究報告 Vol.2010-CSEC-48, No.15 (2010).