

教育用計算機システムにおけるエージェント方式による デュアルブート端末管理

丸山 一貴^{1,a)} 関谷 貴之^{1,b)} 妹川 竜雄^{2,c)} 和田 佳久^{2,d)}

概要: 東京大学情報基盤センターでは 2012 年 3 月に教育用計算機システムを更新し、授業で利用する 1300 台以上の PC 端末について、Mac OS と Windows それぞれのネットワークブート方式の運用から、単一の端末ハードウェアによるローカルかつデュアルブート方式の運用に変更した。端末管理の要件を検討し直した結果、更新後のシステムでは (1) イメージ配信機能、(2) パッチ適用機能、(3) 環境復元機能を必須の管理機能と捉え、それぞれに対応する管理ツールを導入して連携動作させることで要件を満たす管理システムを構築した。一方で、当初想定しなかった問題の発生により、運用での回避を余儀なくされているものもある。本論文では、管理方式を変更した経緯と、管理システムに求めた要件、実現した新たな端末管理システムについて述べる。また、ネットワークブート方式との比較についても述べるとともに、運用上の問題点や解決方法について紹介する。

キーワード: 教育用計算機システム、端末管理、デュアルブート、エージェント方式

Agent Based Management of Dual Boot PC Terminals in Classrooms

Abstract: Information Technology Center, The University of Tokyo replaced Educational Campuswide Computing System (ECCS) in March, 2012. The new ECCS provides 1300+ local and dual boot Macs, on which both Mac OS and Windows are available, instead of distinct network boot terminals for each OS. Since our essential requirements for managing terminals are (1) OS image distribution, (2) patch application, and (3) recovery-by-reboot, ECCS now has three correspondent tools and makes them work together. Some unexpected problems, however, require workarounds in our operation. In this paper, we describe why we switched the management system to agent based one, our requirements to the system, and its details. The comparison with the prior network boot system and our workarounds of some problems are also included.

Keywords: Computer system for education, PC terminal management, dual boot, agent based management.

1. 背景

東京大学情報基盤センターでは教育用計算機システムを構築運用している。同システムは講義等で利用する PC 端末、メールホスティングサービスで使用するメールシステム、これらに付随する認証システム、ファイルサーバ、

プリンタ [1]、ネットワーク装置からなる。当センターでは 2012 年 3 月に同システムを更新し、新システム（以下、ECCS2012）の運用を開始している。

PC 端末へは、授業を担当する教員、学生、運用を担当する管理者からそれぞれ以下のような要望がある。

- 教員
 - Autodesk Inventor, 3ds Max を用いた 3D CAD 及び CG の教育
 - Microsoft Excel で分析ツールを用いた統計の教育
 - Unix 由来のライブラリやツール、操作方法の教育
 - 語学教材を見せながら、LMS 上に用意した問題を利用する講義

¹ 東京大学
Yayoi 2-11-16, Bunkyo, Tokyo 113-8658, Japan
² キヤノン IT ソリューションズ (株)
Mita Avanti, Mita 3-11-28, Minato, Tokyo 108-0073, Japan
a) kazutaka@ecc.u-tokyo.ac.jp
b) sekiya@ecc.u-tokyo.ac.jp
c) imogawa.tatsuo@canon-its.co.jp
d) wada.yoshihisa@canon-its.co.jp

- 学生
 - Windows
 - Office 系ソフトウェア
 - Mac OS
- 管理者
 - 6名の常勤職員*1で、3キャンパスに分散配置された1300台の端末を管理
 - Windows Updateの適用を含む、端末環境の更新
 - 教室を閉鎖せずに端末環境を更新
 - 2種類のOSを提供する場合でも、PC端末のハードウェアを1種類に統一

これらのうち、学生の要望はシステムに関するアンケート [2] で ECCS2012 に必要とされたもののうち、PC 端末に関する上位 3 件である。前システムでは、Unix OS として Mac OS を採用し、Mac 及び Windows それぞれの OS 環境をネットワークブート方式により集中管理していた。教員及び学生の要望に応えながら管理者の要望の多くを満たせる効果的な管理方式であったが、PC 端末のハードウェアは OS ごとに分かれており、管理者の要望が一部満たせない状況にあった。また、Mac OS のブートサーバとして利用してきた Apple Xserve が 2011 年 1 月末で販売終了するとアナウンスされたこともあり、上記の要望に対する回答を改めて考え直すこととした。

要望の中で、Inventor 及び 3ds Max は Windows でしか動作しないアプリケーションであり、また高いグラフィック性能を必要とする。Windows をネイティブ動作とするか仮想マシン環境とするかを検討するため、実際にこれらを講義利用する教員も同席の上で、各環境での動作を検証した。結果として、仮想マシン環境では OS 及び Inventor 等の起動時間が大きく劣っていたため、Windows がネイティブ動作する環境を前提で検討した。その上で Unix 環境を提供する方法として以下の 5 通りがあり得た。

- (1) Cygwin
- (2) Unix OS が動作するサーバ (以下、Unix サーバ) への ssh による、コマンド利用
- (3) Unix サーバと、Windows 上で動作する X サーバソフトウェアの連携による GUI 利用
- (4) Windows 上の仮想マシン環境で動作する Unix OS
- (5) ネイティブ動作する Unix OS

前システムとの互換性を考慮して、ECCS2012 では Unix OS として Mac OS を利用することとし、第 5 項の方式で実現することとした。ハードウェアは Apple iMac であり、Mac OS X 10.7 Lion 及び Microsoft Windows 7 が Boot Camp によりデュアルブートで動作する環境となった。

この構成により、教員と学生の要望はアプリケーション

の性能も含めて満たすことができた。管理者の要望のうちハードウェアの統一は既に述べた通りだが、ECCS2012 は PC 端末として 1321 台の Apple iMac を備え、デュアルブートとしたことから実質的に 2642 台の PC を管理する状況となっており、OS ベンダーから提供される標準的なツール類では他の 3 つを達成することができない。そこで、端末上でエージェントが動作し、サーバ上で設定した指示に従って各端末が自律的に動作して必要なアップデート作業を実施する、エージェント型の管理方式を導入することとし、他の要件を満たす形態を採用した。

本論文では、ECCS2012 で採用したエージェント型の管理方式について、具体的な管理方法や各作業に要する時間を含む運用の実際を説明するとともに、従来採用してきたネットワークブート方式との比較について述べる。

2. ネットワークブート方式の端末管理

2004 年 3 月より運用を開始した ECCS2004、2008 年 3 月より運用を開始した ECCS2008 では、いずれも Mac OS 及び Windows のネットワークブート方式 (以下、ネットブート) を採用し、端末上で動作する OS の集中管理を実現していた。本章では従来採用してきたネットブートについて、その管理方法と長短について述べる。

2.1 ネットブートの仕組み

ネットブートでは、端末の電源が投入されると DHCP により IP アドレスを取得し、セグメント上でネットブートサーバ (以下、ブートサーバ) を検索して接続、ブートサーバから OS イメージが供給され、端末が動作する。1 台のブートサーバが OS イメージを供給する端末の数はブートサーバの性能やネットワーク構成によって異なるが、ECCS2008 では 1 台のブートサーバ当たり約 35 台の端末を接続する構成としていた。ECCS2008 では Mac OS 用に 33 台、Windows 用に 6 台のブートサーバを有しており、これら複数のブートサーバ上にある OS イメージを統一しておくことが必要となる。

図 1 は ECCS2008 における 2 系統のネットブートシステムを示したものである。Mac 端末には専用のブートサーバがあり、OS イメージはブートサーバだけが保持している端末には存在しない。Windows 端末も同様である*2。

OS イメージの更新は、それぞれの OS のマスターイメージを用いて端末を起動し、アプリケーションのインストールやパッチ適用といった変更を施した上で、OS のイメージ化を行い、全ブートサーバに配信することで完了する。

*1 ECCS 全体に加えて、メール及び DNS のホスティング、携帯端末用ネットワーク等の運用とユーザ支援を担当しており、PC 端末の OS メンテナンスだけにかけられる労力は少ない。

*2 Citrix の Provisioning Server によるネットブートに加えて、シー・オー・コンズの ReadCache システムを使用していた。端末上にもキャッシュとしてのコピーは存在したが、ブートサーバ上のイメージが更新されれば破棄される。

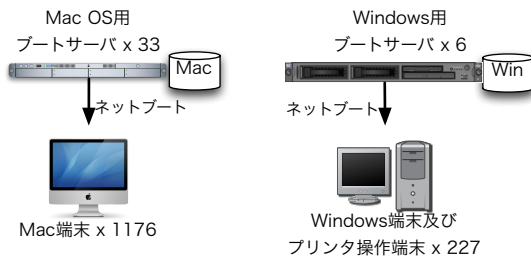


図 1 Mac OS 及び Windows のネットブートシステム
Fig. 1 Network boot systems for Mac OS and Windows.

2.2 ネットブートの長所と短所

ネットブートの長所は、ブートサーバ上の OS イメージを適切に管理することで、動作する端末にインストールされているアプリケーションやパッチのバージョンを完全に制御できるということである。

OS イメージは OS とアプリケーション及び関連ファイルを全て含む単一のファイルであり、環境によって異なるが通常は 10 から 20GB 程度となる。従ってイメージ化のステップでは 10GB 以上のファイルを生成することとなる上に、複数のブートサーバへ配信する必要があるため、相応の時間がかかる。しかし、所要時間の大部分はファイル生成やファイルコピーの時間であり、担当職員は所定の作業を行った上で終了を待つだけで労力がかかるわけではない。ファイル生成を日中に行って生成完了後に動作確認し、夜間にファイル配信を行えば、翌朝には全ての端末が新しい OS イメージで動作することとなる。

一方で短所は、端末台数に応じた数のブートサーバが必要なこと、動作時に若干のタイムラグがあること、ネットブートに固有の問題が起りうることである。

ネットブート端末は、使用状況に応じてオンデマンドにブートサーバから OS イメージの必要箇所を読み出す。例えば動作中にそれまでは使用していなかったアプリケーションを起動すれば、その格納領域をブートサーバから転送することとなり、若干のタイムラグが発生する。また、教育用計算機システムでは 1 教室の全端末が、同時に全く同じ操作をすることがある。そのような場合でも最低限の性能を維持するためには、ネットブート端末とブートサーバの台数比率を極端に高くすることができず、第 2.1 章で述べたような比率とすることが求められる。結果として、端末とは別にブートサーバの導入・運用コストがかかる。

ネットブートシステムに起因する問題や、ネットブートで運用されることを想定していないアプリケーションも存在する。問題の再現性が端末台数や再起動の回数に依存しており、ネットブートに起因する問題であることを特定し、ベンダーに不具合修正を求めて問題が解決するまでに相当の時間を要するケースもあった。

ECCS2004 及び 2008 ではこれらの短所よりは長所を活かして効果的に運用してきたが、Mac OS のネットブート

サーバとなる Xserve が販売終了することから、端末の新たな構成方式を検討することとなった。

3. エージェント方式の端末管理

第 1 章で述べた未解決の 3 つの要望は、(1) 分散配置された端末を多くはない職員でメンテナンスすること、(2) 端末環境の更新、(3) 教室を閉鎖することなく更新することであった。ローカルかつデュアルブートである端末には、OS と講義が必要となるアプリケーションの完全なセットが合計 2 式格納されていることとなり、2012 年 9 月現在では Mac OS が約 30GB、Windows が約 50GB である。

第 1 を解決するためには、ローカルディスクに内蔵されるこれらデータを集中管理して、必要な設定変更や更新を行える必要がある。第 2 を解決するためには、Windows Update を考慮すれば少なくとも月に 1 回の更新作業を行う必要があるが、完全な OS イメージを全端末に毎回配信することは、講義が行われている通常の学期中に第 3 の条件達成が困難となるおそれがあった。従って、個別のパッチ適用やアプリケーション単体のインストールといった、差分のみを更新する仕組みが必要である。加えて、各端末へ個別に更新作業を実施することになると、一部で更新作業に失敗する端末が出ることも考えられる。そのため、更新作業終了後には失敗した端末があるか、ある場合はどの端末かが分かるよう管理者に通知する仕組みが必要と考えた。

一方で、Windows におけるサービスパックに代表される大規模な更新は、差分のみを更新するとしても相当の所要時間が見込まれることや、更新による環境の変化が講義に与える影響を考慮すると学期途中での適用は困難である。従って、講義がない長期休暇期間に、完全な OS イメージの配信によって適用する方が合理的と言える。

また、ローカル起動を前提としたとき、OS 再起動によって利用開始前の状態に戻せることや、全端末の状態が同一であることを保証することで、ネットブートに近い環境を維持したいという管理側の新たな要望が生まれた。このことは我々管理側だけでなく、教員や学生といったシステムの利用者に対しても、従来のシステムと同様の利用感を提供できるというメリットもあると考えた。

以上のことから、新たな端末管理方式に求めた機能は以下のようなものである。

- 管理者の手元にあるマスター OS をイメージ化し、全ての端末に適用して同様に動作させること（以下、イメージ配信）。
- アプリケーションインストールや、OS アップデートの適用といった、差分更新が可能であること（以下、パッチ適用）。また、その結果を通知できること。
- 再起動によって元の状態に復帰すること（以下、環境復元）。

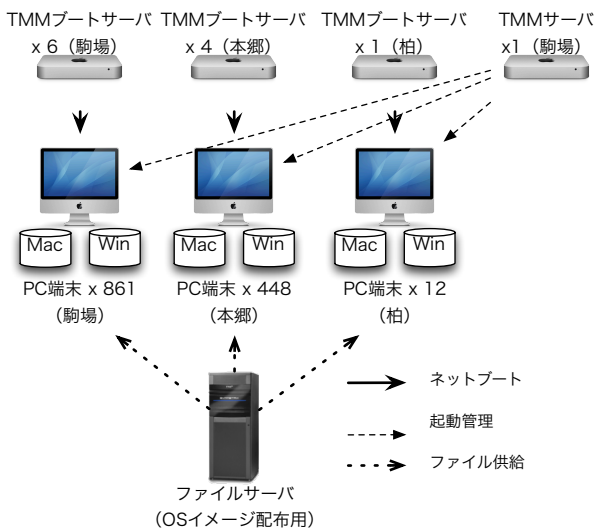


図 2 TMM サーバ群の構成

Fig. 2 TMM servers restore both OSs into PCs.

本章ではこれらをどのように実現したか、システムの具体的な構成について述べる。

なお、イメージ配信及びパッチ適用は利用者がほぼ 0 となる深夜に実施することを前提としている。一部の端末設置箇所は深夜も利用可能であるが、深夜利用はごくわずかであり、第 1 章で述べた教室を閉鎖しないという要件を考慮して、利用者が利用中でこれらが失敗した端末は改めて翌日以降の深夜に実施することとしている。

3.1 イメージ配信

講義用の PC 端末として運用される Mac OS 端末を集中管理するツール群である、キヤノン IT ソリューションズの Total Manager for Mac (以下、TMM) を用いて実現している。TMM は当初、多数のネットブートサーバを持つ環境向けの管理ツールとして開発された [3] が、現在はローカル起動端末のイメージ管理ツールとしての機能も備えている。TMM では管理対象の端末をディスクレスネットブートさせることで内蔵ドライブをメンテナンス可能な状態にし、Mac OS 及び Windows の各パーティションにネットワークを介してイメージ化した OS を書き込むことができる。

図 2 は、TMM によるイメージ配信のために実現したシステム構成を示したものである。PC 端末の両 OS には TMM エージェントが導入されており、起動時及び起動中は TMM サーバと通信する。TMM サーバからは再起動やシャットダウン、起動 OS 指定といった端末の起動管理にかかる指示や、指定時刻に自動起動する設定の投入といった指示を送ることができる。

イメージ配信の手順は以下の通りである。

(1) 事前に、TMM サーバを通じて端末が深夜に自動起動

するよう設定する*3。

- (2) 指定時刻に自動起動した端末は、TMM サーバからの指示に基づいて、イメージ配信のためにディスクレスネットブートする (再起動する)。
- (3) イメージ配信のためにネットブートした端末はファイルサーバをマウントし、そこに配置された OS イメージを読み込んで、所定のパーティションに書き込む。
- (4) 端末がイメージ書き込み後の処理 (コンピュータ名の設定等) を行い、処理結果を TMM サーバに通知し、第 3.3 章で述べる環境復元を有効にして再起動する*4。
- (5) TMM サーバが処理結果を集計して、成否の状況を管理者宛にメール通知する。

イメージ配信では前システムと同様にネットブートを使用しているが、講義利用における場合と異なり、イメージ配信におけるブートサーバへの負荷は小さい。また、ディスクレスネットブート時に端末で RAM ディスクを活用することにより、ブートサーバへの負荷を一層引き下げている。一方で、キャンパス間ネットワークの遅延の影響を受けないよう、TMM ブートサーバは 3 つのキャンパスごとに配置した。これらにより、1321 台の端末に対して合計 11 台のブートサーバという構成 (ブートサーバ 1 台当たり平均約 120 台の端末) であり、なおかつ、ブートサーバを Apple Xserve から Apple Mac mini へ変更したが問題なく動作している*5。

3.2 パッチ適用

パッチ適用はカセヤジャパンの Kaseya[4] を使用して実現した。TMM と同様に、PC 端末の両 OS にエージェントをインストールし、サーバからの指示に基づいて所定の動作を行う。図 3 に、Kaseya のシステム構成を示す。

Kaseya によるパッチ適用は、以下のように TMM による起動管理と組み合わせることで実現する。なお、文中のワークフローとは Kaseya サーバからエージェントに指示する作業ステップを指しており、リストとはエージェントが具体的な適用ファイルを特定するためのデータファイルである。

- (1) 事前に、TMM サーバを通じて端末が深夜に自動起動するよう設定するとともに、Kaseya サーバで該時間帯にパッチ適用のワークフローを設定する。また、適用すべきアプリケーションのインストーラと、これらのリストをファイルサーバ上に配置する*6。
- (2) 指定時刻に自動起動した端末は、パッチ適用対象の OS でない場合、TMM サーバからの指示に基づいて、対

*3 Mac ハードウェアには Wake-on-LAN 機能がないため、代替手段として指定時刻に自動起動する設定を利用している。

*4 配信するイメージは、予め環境復元が解除されている。

*5 Mac OS をネットブートするためには、Mac 系サーバが必要である。

*6 Windows Update は各端末が直接インターネットから取得する。

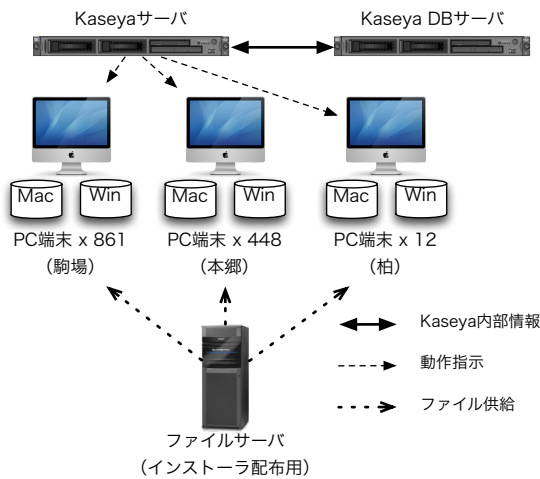


図 3 Kaseya サーバ群の構成

Fig. 3 Kaseya servers make PCs install applications or apply patches.

象 OS に切り換えるため再起動する。

- (3) 端末上の Kaseya エージェントが Kaseya サーバと通信し、処理対象のワークフローがあることを認識して、一連の適用処理を開始する。
- (4) 端末は第 1 項で用意したリストを読み込み、第 3.3 章で述べる環境復元を解除して再起動する。
- (5) リストに従ってアプリケーションのインストールや、必要に応じて再起動を行う*7。
- (6) 端末は適用処理結果をファイルサーバ上のファイルに書き込み、環境復元を有効にして再起動する。
- (7) Kaseya サーバが端末の処理結果ファイルを集計し、成否の状況を管理者宛にメール通知する。

Kaseya エージェントだけでは Mac ハードウェアでの自動起動等を柔軟に制御しにくかったため、TMM を用いて実現している。

3.3 環境復元

ネットブートと同様、再起動によって利用前の状態に環境を戻すこと、全端末の状態が同じであることを保証することを目的として、ECCS2012 では環境復元のためのソフトウェアを利用することとした。Mac OS 及び Windows 双方に対応していることから、Faronics の Deep Freeze を両 OS に導入している。

利用者が端末を利用する間は、ローカルディスク上のファイルは常に Deep Freeze による保護下にあり、ファイルシステムへの変更は再起動により破棄される。管理者は専用のパスワードを用いて、手動あるいは Kaseya エージェントを介して Deep Freeze の解除や再適用が可能である。

*7 適用後に再起動が必要なパッチをインストールする場合は、リストに再起動を指示する命令を入れておく形式で運用している。

4. エージェント方式による端末の運用

第 3 章では、複数の管理ツールを用いて端末管理システム全体をどのように構成するかについて述べた。本章では具体的な更新作業を例にとって、管理者が行う作業手順と更新作業の所要時間について計測した結果を述べる。

4.1 作業例 1：イメージ配信

第 3.1 章のイメージ配信の作業手順について述べる。

最初に、マスターとなる端末の OS をイメージ化する作業が必要となる。まず Kaseya エージェントをアンインストールする*8。Mac OS の場合は必要な更新を施した上で、OS 添付のディスクユーティリティを使用してイメージ化し、該当ファイルをファイルサーバに転送すればよい。Windows の場合はマスター機の OS に必要な更新を施した上で、デフォルトプロファイルの変更やプロファイルのファイルサーバへのコピーを経て、Winclone[5] によるイメージ化を実施する。イメージ化は Mac OS で約 20 分、Windows で約 1 時間を要する。

イメージ配信は TMM のみの作業で実現される。TMM のインターフェースは Web ブラウザを介して提供されるため、管理者は図 2 の TMM サーバにブラウザでアクセスし、管理者アカウントでログインする。次にイメージ配信の対象端末を選択し、対象 OS を指定して、各 OS のイメージファイルを選択する。最後に、今すぐ実行もしくは指定した時間に実行する設定を行えばよい。この作業は慣れれば数分で完了する。

表 1 は、イメージ配信そのものの所要時間を示したものである。高速接続の端末群とは、OS イメージを格納したファイルサーバから 10Gbps を占有するバックボーンで接続された端末群を示しており、低速接続の端末群とは、1Gbps を学内の他システムと共用するバックボーンで接続された端末群を示している。また、TMM で配信するイメージは圧縮しているため、端末上の使用量よりは少ない容量となっている。Windows ではイメージ配信後に自動実行される、コンピュータ名設定等の処理によって、イメージ容量に対する所要時間が Mac OS よりも劣っている。我々の運用では端末の更新作業を深夜 1:00 から朝 8:00 までに実行できればよく、表 1 の各端末群は同時並行で配信できるため、運用上十分な性能を発揮していると言える。

4.2 作業例 2：アプリケーションインストール

第 3.2 章のパッチ適用の具体例として、アプリケーションをインストールする場合の作業手順について述べる (TMM 側の作業については省略する)。

*8 Kaseya エージェントを導入したままでイメージ化して配信すると、Kaseya サーバでは配信後の端末も全てマスター機と同一のものとして判断してしまうため。

表 1 OS イメージ配信の所要時間

Table 1 Time needed to install whole OS image via network.

	高速接続の端末群		低速接続の端末群	
	Mac	Win	Mac	Win
対象 (台)	500	500	315	315
イメージ容量 (GB)	17	25	17	25
所要時間 (分)	50	90	100	170

表 2 Mac OS へのアプリケーションインストール所要時間

Table 2 Time needed to install applications into Mac OS.

	適用例 1	適用例 2
対象 (台)	943	1140
アプリケーション数 (個)	17	3
容量 (GB)	1.6	0.65
所要時間 (分)	180	90

インストール対象アプリケーションのインストーラファイルを取得して、ファイル名やインストールパラメータ等を記入したリストファイル等を作成し、ファイルサーバに保存する。Mac OS の場合は、pkg 形式のインストーラを有するか、app 形式のファイルを Applications フォルダにコピーすればよいタイプのアプリケーションがほとんどであり、Kaseya によるインストールが容易である。Windows の場合はサイレントインストール^{*9}に対応したアプリケーションは容易にインストールできるが、対応していない場合はパッケージャを用いて適用することになる。

パッケージャとはインストールの前後でファイルシステムの差分を取るツールであり、インストールによって施される変更を記録し、適用対象の端末に対して全く同じような変更を適用する。Windows アプリケーションではサイレントインストールに対応していないことも多く、パッケージャによるインストーラ作成が有効な手段となっている。

表 2 は Mac OS へのアプリケーションインストールに要した時間を示したものである。第 4.1 章でも述べた通り、運用上の問題は無いと言える。

4.3 作業例 3: Windows Update 適用

Windows Update は対応するパッチファイルを実行するものであり、アプリケーションインストールと同等の操作であるが、月例で判断が必要となる重要な処理であることから、Kaseya 上でも特別に専用の処理手順が存在する。各端末上のエージェントが適用済みのパッチを検査して図 2 の Kaseya サーバに送信しており、各端末ごとに未適用のパッチが管理された状態になっている^{*10}。

Microsoft によって公開されたパッチを適用するため、管

^{*9} ウィザードやダイアログのようなユーザ操作を必要としないインストールモード。

^{*10} ECCS2012 では、端末ごとに環境が異なることと講義利用への影響が懸念されることから、端末によるバラツキがない状態にすることを前提としている。従って、原則として全ての端末において未適用パッチの状況は同一である。

表 3 Windows Update の所要時間

Table 3 Time needed to apply Windows Update.

対象 (台)	587
パッチ数 (個)	13
容量 (MB)	47.2
所要時間 (分)	120
失敗端末 (台)	16

理者は Kaseya サーバにログインして適用すべきパッチを「承認済みパッチ」として選択する。続いて、承認済みパッチの適用対象となる端末を選択する。これらの設定内容をファイルとして保存し、Kaseya サーバ上に配置する。続いて TMM で Windows 起動スケジュールを設定し、Kaseya でパッチ適用に関するいくつかのアクションと、更新後の集計ワークフローを設定して終了となる。

表 3 は、一部の端末に Windows Update を実施した際にかかった時間を計測したものである。これまでの作業例と同様に所要時間は許容できるものであるが、16 台が適用に失敗した。具体的な原因は第 6 章で述べるが、該当端末が指定時刻に起動しなかったことに起因している。これらを翌日の日中に利用者が起動して使用すればその夜には自動起動が行われるため、Kaseya の Windows Update 適用フローを 1 晩だけでなく連続した数日間の夜間に設定することで、16 台のうち 14 台については後日適用が完了した。

5. 2 種類の管理方式の比較

本章では、ネットブート方式とエージェント方式について、3 つの観点から比較を行う。

5.1 ハードウェア構成

ネットブート方式では端末 1403 台あたりにブートサーバ 39 台を設置していたが、ECCS2012 ではイメージ配信用のみ設置しており、端末 1321 台あたりにブートサーバ 11 台となっている。前システムと同じ割合でブートサーバを設置する場合には約 36 台が必要となるが、実際には 11 台でありごく小規模になっている。

以前のブートサーバは全て 1U サーバであり、合計 39U を消費していた。ECCS2012 では Apple Mac mini を使用しており、棚板を用いて 1U あたり 4 台を設置^{*11}できるため合計 3U で設置可能である^{*12}。管理サーバは、TMM サーバ 1 台 (1U サーバ) から、TMM サーバ 1 台 (Apple Mac mini) と Kaseya サーバ 1 台 (仮想マシン) 及び Kaseya DB サーバ 1 台 (1U サーバ) に変更された^{*13}。ブートサーバの設置面積や消費電力は大きく削減できたとと言えるが、

^{*11} ラック前面と背面に、横並びで 2 台ずつ設置できる。

^{*12} 実際には 3 キャンパスに分かれて設置されており、合計 5U を消費している。

^{*13} 前システムにおける NEC の System Management for Advanced netboot はネットブートサーバ上で動作していたため、ここでは数えない。

一方で更新作業時には全端末が夜間に起動して動作するので、端末システム全体として省電力を達成できたかは未確認である。

5.2 管理ツール

前システムから引き続き利用しているものは TMM であり、内部では Mac OS X Server に標準で添付されているネットブートサービス及びネットワークインストール機能を利用している。Windows のイメージ配信を行うため、ECCS2012 からは TMM 内部で Winclone が新規に追加されている。Windows を提供するために、前システムではネットブートを利用しており、Citrix の Provisioning Server 及び NEC の System Management for Advanced netboot から構成されていた。ECCS2012 では Mac OS に標準添付される Boot Camp を利用するよう置き換わっている。パッチ適用のための Kaseya 及び環境復元のための Deep Freeze も、ECCS2012 から新たに利用している。

管理者は、前システムでは OS ごとに独立した管理ツールを操作していたが、ECCS2012 では密接に連係動作する 2 種類の管理ツールを操作することとなる。要件を完全に満たす単一の管理ツールがない場合でも、2 つのツールを相補的に動作させることで、少人数で集中管理する構成の実現に成功した。

5.3 更新作業

第 2 章で述べた通り、従来は端末の更新を行うたびにイメージ化と全ブートサーバへの配信という作業が必要だったため、基本的にはセキュリティパッチが出た際などにまとめて更新を行っていた。ECCS2012 ではイメージ配信を行う際、同様にイメージ化が必要となる。また、配信先は全ブートサーバから全端末に変わっており、配信時間と結果確認はより時間がかかるものになっているが、第 4.1 章で述べた通り夜間に実施可能であり、従来と同様の運用が可能である。

パッチ適用による更新作業はイメージ化が不要であることに加えて、全端末に転送すべきデータはパッチやインストーラのみであるから OS イメージ全体よりは少ない。一方で、更新の前後には環境復元の解除と再適用が必要であり、都度再起動を実施する。Windows Update の前後ではパッチ適用状況のスキャンを行い、パッチやインストーラによっては途中で OS 再起動も要求されることから、OS イメージをシーケンシャルに転送するよりは所要時間が長くなると言える。更新作業は利用者がほぼ 0 となる夜間に限って実施するため、前システムで実施していたようにまとめてパッチを適用する方法よりは、細かな更新をその都度適用する方が有効であると考えられる。

1 回の更新作業に必要な手順は、ネットブートではマスター機を用いて作業し、結果のデータを全ブートサーバに

配信するだけでよかったのに比べ、インストーラの形式に合わせてインストール方式を選択したり、複数の管理ツールで更新設定を行う必要があるなど、作業の手間自体は増加していると言える。しかしながら、第 4.2 章及び第 4.3 章で述べたように、パッチ適用に必要な作業のステップ数は多いが、慣れてしまえば機械的な作業であるとも言える。パッチ適用設定作業への習熟度を高めることで、より迅速な更新が可能になると考えている。

6. 既知の問題

ECCS2012 では、端末が自律的に動作して内蔵する 2 つの OS を更新するシステムを設計した。従って、端末を遠隔（管理システム）から制御できることが前提であり、各管理ツールのエージェントを導入することでこれを可能にしているが、当初計画した通りには動作していない部分もある。本章ではこうした既知の問題と、講じている回避策について述べる。

6.1 夜間の自動起動

OS 更新作業のために全端末を夜間に起動する必要がある。一般には Wake-on-LAN と呼ばれる仕組みによって実現されるが、ECCS2012 で導入している Apple iMac はこれに対応していない。代替手段として、指定時刻に起動する設定を事前に施すことでこれを実現している^{*14}。

問題は、Windows を起動した際に発生する。Boot Camp の仕様により、Windows 起動時にはこの起動時刻設定が削除されてしまう^{*15}。従って、Windows を起動してそのままシャットダウンしてしまうと、その端末は自動起動することができず、必要な更新作業が実施されないことになる。第 4.3 章で述べた Windows Update の失敗端末とは、このような理由で起動してこなかったものが該当している。

対策として、Windows にログインした場合は利用終了時に再起動しか選択できない設定として、その後で Mac OS が起動する（起動時刻設定がなされる）ようにしているが、以下のような場合には夜間の自動起動が行われない。

- Windows のログイン画面でシャットダウンを選択
- Windows 起動中の、電源ボタン長押しによる強制シャットダウン
- 利用者が端末の画面角度等を調整した際に電源ケーブルの接続が不十分になり、通電しなくなった場合

このような状況になっても、多くは翌日以降に利用者が起動することで起動時刻設定がなされるため、担当者が設置場所まで出向いて対処するケースは少ない。

^{*14} 具体的には Mac OS 上の pmset コマンドを使用する。iMac は電源投入後に Mac OS が起動し、BootPicker による OS 選択画面が現れる設定となっている。このときに TMM サーバからの指令により、夜間の自動起動が設定される

^{*15} ハードウェアクロックのタイムゾーンが判断できないため、意図しない時刻に起動しないよう、このような仕様になっている。

6.2 イメージ配信後の端末登録

イメージ配信後に Kaseya エージェントを再度インストールする処理が実行される。その際 Kaseya エージェントは新たな GUID を生成して Kaseya サーバと通信するため、サーバ側では新しい端末が出現したと判断され、配信前の端末と同一であるとは認識できない。結果として、Kaseya サーバ上では新旧の端末（新旧の OS）が二重に登録された状態となってしまう。現在はこれらを統合する作業を Kaseya サーバの Web 画面から行っているが、改善についてカセヤジャパンを含めて検討している。

7. 関連研究

丸山ら [6] は Windows と Unix を仮想マシンにより共存させる場合の、ホームディレクトリの提供手法やゲスト OS の起動時間短縮について論じている。中澤ら [7] は Windows 端末をネットブートし、その上で仮想マシンにより VineLinux を用いた Unix 環境を提供している。また、Windows 環境の Cygwin、Windows 環境の X サーバソフトウェアと外部 Unix サーバが連携する仕組みも提供しており、第 1 章で述べた Unix 提供方法のうち 3 項目を実現している。ECCS2008 では、Mac OS をネットブートさせた上で仮想マシンによる Windows 環境の提供も行っていたが、講義利用において Unix と Windows の同時利用が必須ではなかったこと、ネイティブ動作の方が性能上望ましかったことから、ECCS2012 では仮想マシンによる方式を採用していない。

Palomino[8] は、Radmin を用いたデュアルブート Mac による Reed 大学端末室の OS メンテナンスについて報告している。Radmin は Mac OS を含む Unix 上で動作するツールで、マスターとなる OS イメージと各端末のディスクをファイル単位で比較して、マスター側の変更を反映する。Reed 大学では環境復元ツールを利用しておらず、この機能を毎日夜間に動作させることで、日中の利用で施されたファイルシステムへの変更を全て破棄している。夜間に端末を起動して更新処理を実施する方法は本論文で述べた方式と同様であり、また、Radmin のファイル比較処理は第 4.2 章で述べた Kaseya のパッケージによる方法と同等と言える。一方で、この方式では Windows 起動時にハードウェアが追加されたと認識して再起動を促されることが報告されている。ECCS2012 ではエージェント方式によって Windows 自身がインストーラを実行することで、このような問題が発生しない環境を構築している。

8. まとめと今後の課題

本論文では 1300 台を超える教育用計算機システムの端末において、Windows 及び Unix OS を提供する方法として、デュアルブート端末による環境提供とエージェント方式による集中管理について、検討の過程と具体的な実現

方式について述べた。OS の更新方法にはイメージ配信とパッチ適用の 2 種類を準備し、それぞれ大規模な変更をまとめて適用する方法と、細かな差分更新を個別に実施する方法として位置づけている。加えて環境復元ツールを導入することにより、再起動で端末の環境を元に戻すというネットブートに近い環境を継続した。2 種類の更新方法を用いて、実際に夜間実施した場合の所要時間を示し、運用上影響のない処理時間で完了していることを述べた。

今後の課題の 1 つに、故障端末交換時の対応がある。故障端末を交換した場合には内蔵ディスクの内容を他の端末と同一にする必要があるが、イメージ配信を行った上で、その後に実施したパッチ適用を追加実施する必要があり、作業のステップが多く複雑である。パッチ適用のたびにイメージ化する方法もあり得るが、細かなパッチ適用のたびに OS 全体をイメージ化するのは、イメージ化の作業ステップが比較的多いことから望ましくない。そこで、利用されていない端末を TMM から指定することで、自動的にその端末の内蔵ディスクをイメージ化する仕組みを実現することを検討している。

謝辞 本システムの構築及び運用にご協力いただいている日本電気株式会社、カセヤジャパン株式会社の方々に感謝します。

参考文献

- [1] 丸山 一貴, 関谷 貴之: 学外システム連携による教育用計算機システムプリントサービス, 情報処理学会研究報告, Vol.2012-IOT-16, No.9(2012).
- [2] 東京大学情報基盤センター: ECCS に関するアンケート, 入手先 (http://www.ecc.utokyo.ac.jp/files/2011/03/28/ECCS_questionnaire_2010-11-15.pdf)(2012.09.13).
- [3] 安部 達巳, 下村 健, 妹川 竜雄, 関谷 貴之, 丸山 一貴: 運用の省力化のためのシンクライアント管理ツールの導入と評価, 情報処理学会研究報告, Vol.2009-IOT-004, pp.79-84, 2009.
- [4] カセヤジャパン: Automated IT Management Software Solutions | Kaseya, 入手先 (<http://www.kaseya.jp/solutions.aspx>)(2012.09.13).
- [5] twocanoes software: twocanoes software | Winclone 3.4, 入手先 (<http://twocanoes.com/winclone/>)(2012.09.13).
- [6] 丸山 伸, 最田 健一, 小塚 真啓, 石橋 由子, 池田 心, 森幹彦, 喜多 一: Virtual Machine を活用した大規模教育用計算機システムの構築技術と考察, 情報処理学会論文誌, Vol.46, No.4, pp.949-964(2005).
- [7] 中澤 篤志, 間下 以太, 清川 清, 竹村, 治雄: 大阪大学の情報教育システム, 情報処理学会研究報告, Vol.2012-CLE-7, No.8(2012).
- [8] Tony Palomino: Reliable Management of Dual-Boot Labs Using Radmin, Proceedings of the 38th annual fall conference on SIGUCCS(SIGUCCS '10), pp.281-283(2010).