

端末教室における環境統一と端末の高速性とを 両立する管理ソリューション

丸山 伸^{1,a)} 大平 健司^{2,b)} 佐野 雅彦² 谷岡 広樹² 松浦 健二² 上田 哲史²

概要：端末教室における環境統一にはネットブート技術が適している。このネットブート技術を利用すると端末の一斉起動や一斉操作でサーバーやネットワークに負荷集中が生じるが、端末側にディスクイメージの複製を持つことで負荷集中を回避できる。しかしながら、各端末にディスクイメージの複製を持つように設計すると、ディスクイメージを更新した際に各端末が持つ複製が同期されるまでは負荷集中が生じがちとなり端末の高速性が失われることが課題となっていた。

そこで本研究では、端末の高速化に必要な最小限の領域のみを複製することでディスクイメージ更新後における端末側への同期処理を最小化し、ネットブートによる環境統一と端末の高速性を常に両立する手法を提案する。また、徳島大学の端末教室において提案手法の評価をおこない、本提案が有効であることを確認した。

Management solution of PC classroom to balance unification of desktops and terminal's speed

SHIN MARUYAMA^{1,a)} KENJI OHIRA^{2,b)} MASAHIKO SANNO² HIROKI TANIOKA² KENJI MATSUURA²
TETSUSHI UETA²

1. はじめに

端末教室においては多数の学生に対する講義を効率よく実施するために、多数の端末を環境統一することが求められる。また多数の学生で少数の端末を共有する環境のため、前の学生の作業の影響を排除し情報の漏洩を防ぐ観点から、端末を再起動するだけで統一された環境の初期状態に復元する機能が求められる。その一方で、昨今のセキュリティ脅威の増大に伴いセキュリティ更新の頻繁な実施が求められていることや、教育環境の改善のためにアプリケーションを追加したり設定を変更したりする要望が随時寄せられる。このことから、端末教室においては、「環境を統一する」「迅速に再起動できる」「頻繁に環境更新でき

る」ことが求められる。

環境統一を目的とする管理ソリューションとして「ディスク配信技術」と「ネットブート技術」とが広く知られている。そのうちディスク配信技術は、環境更新を行う度に配信作業に時間がかかるだけでなく、配信後の各端末での初期化作業が確実に完了したかどうかを各端末において確認する必要がある。そのため環境更新の作業コストが大きく更新頻度を上げられない。ネットブート技術の持つ「サーバー側での設定変更のみで複数の環境を迅速に切り替え出来る」という特徴は、多様な目的に利用される端末教室の運用に適するが、端末を一斉に再起動した際に負荷集中に伴う性能劣化が発生しやすい欠点をもつ。また、ネットブートにおける負荷集中を避けるために、ディスクイメージ全体の複製をキャッシュとして端末側に持つ構成も考えられるが、この構成では事実上ディスク配信技術と同等の構成となってしまうネットブートとしての特徴が失われてしまう。

そこで著者らはネットブートを基盤としつつ、ディスク

¹ 株式会社シー・オー・コンヴ / CO-CONV, Corp.

² 徳島大学情報センター
Center for Administration of Information Technology,
Tokushima University

a) marushin@co-conv.jp

b) ohira@tokushima-u.ac.jp

イメージ全体の複製を各端末に保持しようとするのではなく、サーバーやネットワークへの負荷集中を避けることを目的として少量のキャッシュを各端末に保持することを提案してきた。さらにキャッシュの構造を最適化することで端末起動が高速になることも示している。[1] また同著者はブートローダ段階におけるキャッシュ技術も利用しつつ、Windows の起動時のディスクアクセスパターンを分析することで、起動時のトラフィックを削減し、端末の起動を高速にする手法も提案してきた。[2] これらの手法の組み合わせにより端末が起動（再起動）に要する時間を最小限にできる。

しかしながら、これらの研究の成果はいずれも端末側に適切な構成のキャッシュが蓄積された状況での端末の起動時間の短縮に関する研究であり、ディスクイメージの更新に伴う端末の起動時間への影響については検討されていなかった。ディスクイメージを更新すると、端末内に蓄積されたキャッシュの全部ないし一部が利用できなくなるため、これまでの評価結果をそのままに適用できない。

そこで、本研究ではディスクイメージ更新時の挙動に着目し、管理ソリューションの再検討をおこなう。まず、「ディスク配信技術」「ネットブート技術」「ネットブート技術をベースにディスク全体をキャッシュする」「ネットブート技術をベースに適切な量をキャッシュする」の4つの技術についてディスクイメージ更新の影響を比較する。また端末教室で多く利用されている Windows 環境について、ディスクイメージ更新時に作成される差分データの内容と端末起動時のアクセスとを分析し、更新作業を実施した際に生じる差分データのうち、端末の起動時に必要となる量は 100~200MB 程度であることを確認した。この結果を元に、ディスクイメージが更新された際の挙動においても、常時サーバーとの通信を行える環境においては「ネットブート技術をベースに適切な量をキャッシュする」方式が端末教室での利用に適した管理ソリューションであることを確認した。

以下、2章では従来手法の分析を行い、3章で本研究の提案を説明する。4章で提案する方針について分析と理論的な評価を行い、5章で実装について述べる。また提案手法を徳島大学の端末教室で評価した結果について6章で述べる。

2. 従来手法

端末教室で利用される環境統一を目的とした技術には「ディスク配信技術」と「ネットブート技術」とがある。^{*1}

^{*1} サーバー上に配置された共通環境を多数の端末から遠隔で操作する VDI 技術も端末教室としての利用が検討されることもあるが、VDI 技術はサーバー側にリソースを集中させる構成のため、激しい負荷集中が生じる端末教室としての利用との親和性が低い。また、VDI においてもサーバー側における環境統一のために「ネットブート技術」ないしは「ディスク配信技術」が用いられるもの

また、ネットブートを基盤としつつ、端末側にディスクイメージの複製を持つキャッシュ技術が併用されることも多い。ここでは、「ディスク配信技術」「ネットブート技術」「ネットブート技術をベースにディスク全体をキャッシュする」「ネットブート技術をベースに適切な量をキャッシュする」の4種類についてそれぞれの技術背景と特徴を述べ、次いでディスクイメージ更新を行った際の影響について述べる。

2.1 ディスク配信技術

ディスク配信技術では、初期状態として作成されたディスクイメージを各端末に複製し環境を統一する。また、端末が起動する度に複製された初期状態から起動するように書き込みキャッシュ技術を用いて環境が変化しないようにする。この技術ではディスクイメージを端末に複製する作業時のみサーバーやネットワークに負荷がかかるが、その後の端末の運用においては負荷集中の問題は生じない。さらに、この複製にマルチキャストやブロードキャストを使えば、配信時のサーバーやネットワークへの負荷も最小限に抑えられる。

しかし、ディスク配信技術では、ディスクイメージを更新する度に更新されたディスク全体ないしは端末側に配信済みのディスクとの差分を各端末に転送する必要がある。また、配信には相応の時間がかかるものであるため、更新したディスクを運用に供するまでに時間を要する。さらには、配信後に各端末で実施される初期化処理を失敗することもあるため、配信後には各端末の動作確認を要するなど管理コストが高い。また、配信作業を行っている間は端末の運用を停止せざるを得ないという制約もある。

2.2 ネットブート技術

ネットブート技術では、サーバー上に配置された初期状態のディスクイメージをネットワーク越しに配信しながら端末を起動する。多数の端末で1つのディスクイメージを共有するため、多数の端末を同じ環境に揃える目的に利用しやすい。各端末は起動する度にサーバー上のディスクイメージの状態から起動するため、環境が変化することはない。また、ネットブートには「サーバー上のディスクイメージを更新して多数の端末に割り付け、各端末が次に起動するときには更新されたディスクイメージで起動する」や「複数のディスクイメージを端末に割り付けておき、講義の内容に応じて端末起動時に選択させる運用を容易にできる」といった特徴もあるが、いずれも端末教室での利用に適している。

しかし、ネットブートは各端末が起動時や稼働中に発生させるディスクアクセスをすべてサーバーと通信を行い処

でもするため、本稿では VDI 技術には触れないものとする。

理するため、端末の台数が多くかつ講義の開始時終了時に多数の端末が一斉に起動・終了される端末教室では、サーバーやネットワークへの負荷集中が発生し性能劣化することが問題となる。負荷に耐えられるようなサーバー台数とネットワーク構成としたり、各教室にサーバーを設置する方針も検討できるが、サーバーの台数が増えるとサーバーの管理コストの問題が懸念される。

なお、ディスクイメージの更新を行った際には、サーバー上で更新されたファイルがメモリ上にキャッシュされているかどうかの差はあるものの、端末の性能への影響はほとんど無い。

2.3 ネットブート技術をベースにディスク全体をキャッシュする

ネットブートにおいては、端末を一斉起動するなどサーバーやネットワークに負荷が集中した際に、性能が低下する。特に多数の端末を一斉に起動した際に起動時間が延びることは運用上大きな課題となる。そのため、端末側にディスクイメージ全体をキャッシュしておき、端末起動時など負荷集中する際に活用して転送量を減らす方式が利用されている。ディスク配信技術と同様にディスクイメージ全体の複製を持つことになるが、ネットブート技術と併用することで現行のディスクイメージで起動している間に更新されたディスクイメージをキャッシュする事ができ、配信のために運用を停止する必要がない。さらに端末側にディスクイメージが保持されることで、サーバーへの接続が出来ない状態でも端末を起動できるメリットが生じる。

しかしながら、この方式は端末側にディスクイメージ全体を保持しつづけることが必要のため、ディスクイメージ更新をするたびにディスクイメージに生じた差分の全体を各端末に配信する必要が生じる。また、この配信が完了するまでは更新されたディスクを利用できないことも運用における制約となる。さらには更新により生じた差分のサイズが大きくなると、配信にかかる時間が長くなるという懸念も生じる。配信すべき差分のサイズについては3章で検討する。

2.4 ネットブート技術をベースに適切な量をキャッシュする

ネットブートにおける負荷集中を避けるためにキャッシュを持つのは前項と同じであるが、ディスクイメージ全体をキャッシュしようとするのではなく、「起動直後にアクセスのある箇所のみ」「頻繁に使われるアプリケーションを起動したとき」など負荷が集中するときのリクエストのみをキャッシュする手法も提案されている。八田 [1] はネットブート環境においてはディスク全体をキャッシュするよりも、マッピングテーブルの機構を用いてリクエストのあった順にキャッシュ領域の先頭からキャッシュデータを

貯めていく構造が好ましいとしている。また、キャッシュデータにおける連続性をキャッシュデバイス上での連続性に対応させるため、キャッシュ領域は専用のパーティションに保持するのが好ましく端末起動時間の短縮に効果的であると述べている。この研究においてはキャッシュデバイスはハードディスクドライブを用いて評価をしているが、キャッシュデータがデバイス上で連続しているとキャッシュデータへのアクセスにおけるリクエスト数の削減につなげることが出来るため、SSD を用いている場合においても効果的である。また、丸山 [2] はブートロード段階のデータもキャッシュすることにより、端末起動時の転送量を最小限にするとともに起動時間をさらに短縮している。

この方式を用いつつディスクイメージを更新した際の影響についても3章で検討する。

3. 本研究の提案

ディスクイメージを更新した際の影響を分析するため、まずは更新により生じる差分ディスクについて分析する。ここでは、端末教室において主に用いられている Windows OS 環境について検討する。

3.1 差分ディスクのサイズとデータ量

ディスクイメージを更新すると、更新作業時にディスクに対して書き込まれた内容が差分ディスクとして蓄積される。更新作業は「アプリケーションの更新」「設定の変更」「セキュリティの更新」などを目的として行われるものであるが、この更新された差分には「実際に更新作業の目的としていた内容」以外に、「更新作業のために端末を起動した際に作成・消去された一時ファイル」と「サーバー上で管理される差分ディスクの構造に起因する無駄な領域」とが含まれる。

まず、Windows が利用するファイルシステムである NTFS は、ディスク上のデータを 4KB サイズのクラスターを単位として管理し、ディスクからの読み込み・書き込みの処理は原則として 4KB を単位として行われる。そのため、書き込むデータが 1 バイトであったとしてもディスク上には少なくとも 4 KB の書き込みが生じる。

つぎに、サーバー側で差分ディスクを保持する際には、vhd 形式や vhdx 形式 [3] が用いられる事が多い。これらのフォーマットでは「ディスクイメージ全体をそのままのサイズで保持する」「ディスクにデータが書き込まれている場所のみを保持する」の2種類の形式が定義されている。仮想ディスクを前者の形式で作成するとデータの書かれていない場所も含めた非常に大きなサイズのファイルとなるため扱いにくい。一方、後者の形式ではブロックと呼ぶ数 MB 単位の単位でディスクを管理するため、仮に 1 セクタ (=512 バイト) のみの書き込みでも 1 ブロックのサイズを消費することになる。

本稿においてはこの先サーバー上での差分ディスクのサイズのことを「ファイルサイズベースの容量」、実際にデータの書かれている容量のことを「コンテンツベースの容量」と呼ぶことにする。社内での評価において、単に端末を起動し即座に終了した場合においても、前者のサイズはおよそ 500MB、後者のサイズはおよそ 200MB となる。大規模な Windows Update を実施した際で、前者のサイズは 20~30GB、後者のサイズでも 1GB 以上となる。

3.2 ディスク全体をキャッシュするディスク更新の影響

ディスク全体をキャッシュし続ける方式の場合には、ディスクイメージ更新後に差分ディスク全体を各端末に配信する必要がある。この場合には差分ディスクの少なくともコンテンツベースの容量の転送を要することになる。サーバー側では「実際に更新作業の目的としていた内容」と「更新作業のために端末を起動した際に作成・消去された一時ファイル」とを区別できないため、実際に端末側で必要とされる差分以外のデータも配信することとなり転送量が増える。すなわち、各端末に配信すべきデータのサイズは 1GB 以上になる可能性があり、端末の利用中にバックグラウンドで配信したり、マルチキャストを使って配信するなどの工夫を行ったとしても、多数の端末に配信するには時間がかかる。

3.3 適切な量をキャッシュする場合のディスク更新の影響

端末が実際に要求した部分のみをキャッシュする方式の場合には、ディスクイメージを更新したとしても実際に更新された箇所にアクセスしない限りは転送は発生しないと想定される。特に、負荷集中が起きがちな端末起動時に読み込まれるモジュールが更新される可能性は高いことや、もしこれらが更新されていたとしても転送を要するのは更新されたファイルのコンテンツベースの容量となり、多数の端末が一斉に転送したとしても負荷集中は起きにくいと想定される。

3.4 提案する構成

「ネットブートをベースとしつつ適切な量をキャッシュする方式」が、ディスクイメージを更新した際の影響の比較においても、負荷集中が発生しづらいこと、イメージ更新直後に端末起動に利用出来るといった点で、端末教室で利用するソリューションに適すと考えられる。

サーバー側では vhd 形式ないしは vhdx 形式でデータを保持するのが適すと考える。多くのディスク容量を消費することにはなるが、端末が起動する際の転送処理にはサイズの影響は出にくいことに加え、標準的かつ汎用的なディスクフォーマットを採用することが運用コストの削減につながると考える。

端末側では、「キャッシュ容量はそれほど多くは必要とし

表 1 試験端末 70 台

Table 1 Device specs

筐体:	Fujitsu Esprimo K556/M
CPU:	Core i3-6100T
Memory:	8GB
Disk:	Samsung MZ7TY128HDHP 128GB
NIC:	Realtek GBE Ethernet RTL 8168
OS:	Windows 10 Enterprise 2016 LTSC
Firmware:	BIOS モード

表 2 サーバー 4 台

Table 2 Server specs

筐体:	Fujitsu BX2560 M2
CPU:	Xeon E5-2698v4 (仮想) 6C/6T 2.20GHz
Memory:	24GB
Disk:	(仮想) システム用 60GB 仮想ディスク用 4.5TB
NIC:	(仮想) vmxnet3
OS:	Windows Server 2012 R2
仮想化基盤:	vSphere Hypervisor

ない」こととディスクアクセスの多くは 4KB を単位として読み書きが行われるため、端末側に保持するキャッシュは 4KB 単位で管理するのが好ましいと考えられる。

4. 試験と評価

徳島大学情報センター（学生数 7525 名）では 2017 年 2 月にシステム更新を行った。更新後のシステムは Windows 端末 681 台をネットブートで稼働させる構成とした。このシステムに対して本提案の管理ソリューションを導入して約半年間の運用を行った。

端末は 2 つのキャンパスに分散して配置されている。別拠点のデータセンターに仮想基盤が構成され、そこに 4 台のネットブートサーバと DHCP サーバおよび TFTP サーバが設置された。

各キャンパスの端末とデータセンター設置のサーバとの間は十分に広い帯域で結ばれており、かつ遅延も十分に小さいと見込まれたため、各キャンパスには DHCP サーバ、TFTP サーバなども含めてサーバは一切配置されていない。

4.1 試験環境

評価試験を実施した環境の機器構成は表 1 及び表 2 の通りである。ネットブート基盤として表 3 を利用した。また、ネットワーク構成は図 1 の通りである。

4.2 ディスクイメージについて

ディスクイメージは全部で 22 種作成され、そのうち主に 5 種類が利用されている。以下では、これらを代表して、

表 3 ネットブート基盤
Table 3 Netboot software

Citrix Provisioning Services 7.11
CO-CONV ReadCache 4.6.1.0
CO-CONV CO-Store 4.5.1.4
CO-CONV CO-Booster

表 4 キャッシュ有効時のトラフィック
Table 4 Traffic of netboot with cache

	Request	Response
段階 1a	1647 個, 99 Kbyte	1627 個, 965 Kbyte
段階 1b	231 個, 25 Kbyte	228 個, 293 Kbyte
段階 2	1741 個, 167 Kbyte	1838 個, 333 Kbyte

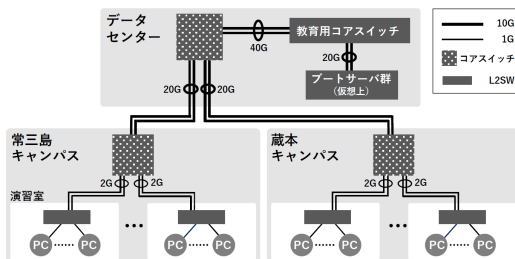


図 1 ネットワーク構成図
Fig. 1 Network layout

“common1”と呼ばれているディスクについて述べる。
 端末に提供している C ドライブサイズは 150GB. 現時点で 55.6GB が消費されている. 導入されているアプリケーションは, Office, Tex, Emacs, gvim, Java, Firefox, Chrome,R など, 多岐にわたる. ウィルス対策・標的型攻撃対策ソフトとして Symantec Endpoint Protection と yarai が導入されている.

この運用イメージに対しては, 自動的に毎週 1 回 Windows Update とウィルス対策パターンの更新が実施されている. その他, アプリケーションの更新作業なども随時実施されている. そのため運用期間は半年であるがその間に作成されたバージョン数は 64 となっており, バージョン 16 以降が残されていてこのディスクは全バージョンで 469.5 GB を消費している.

4.3 端末起動に要する時間

端末起動に要する時間を評価するため, 端末 68 台の電源投入から自動ログオンを経てデスクトップが表示されるまでの時間を計測した.

4.3.1 キャッシュなしでの一斉起動

キャッシュがまったくない状態からの端末 68 台の一斉起動でログオン画面が表示されるまでに各端末は 1.4GB の読み込みを行っており, 要した時間は約 6~7 分であった. このとき, サーバーの CPU 負荷は 15~20%, ネットワークの出力は約 500Mbps であった. サーバーが 4 台に負荷が均等に分散されていることから, サーバーの CPU やディスクではなく, 教室の上流ネットワークがボトルネックになっていることが確認できた.

4.3.2 キャッシュありでの一斉起動

端末にキャッシュが蓄積された状態で端末を起動すると, 端末 1 台のみを起動の際で 56 秒, 68 台の一斉起動でも

表 5 内蔵 SSD の読み込み性能
Table 5 Read performance of SSD

Sequential Read (Q= 32,T= 1) :	550.959 MB/s
Random Read 4KiB (Q= 32,T= 1) :	346.765 MB/s
Sequential Read (T= 1) :	526.152 MB/s
Random Read 4KiB (Q= 1,T= 1) :	42.762 MB/s

64 秒という高速な結果が得られた. 本案件ではサーバーを DC に設置しているが, 通常の運用時には端末側のキャッシュが十分に効いていて転送量もリクエスト数も少ないため, 一斉起動をした場合にも起動時間への影響は小さい.

この状態での端末起動時の転送量を表 4 に示す. ネットブートであってもキャッシュを有効にすればサーバとの通信は極めて少ない. TFTP での転送量 (段階 1a のバイト数が多いのは, ブートメニューの背景画像の転送によるものである.*2

起動が速い理由としては, 「デフォルトプロファイルのサイズが小さい」ことと「内蔵 SSD が高速」であることがあげられる. まず, 固定ユーザプロファイル環境で Default プロファイルのサイズは 60.1MB (ファイル数 1765 個, フォルダ 1000 個), そのうち AppData が 54.1MB とコンパクトであった. 次に Crystal DiskMark Ver.5.2.2 で端末の内蔵ディスクの性能を計測した結果を表 5 に示す.

IOPS が課題となる OS の起動初期 (段階 1b) においては Random Read 4KiB (Q1T1) が, キャッシュデータを一気にメモリに読み込むキャッシュ初期化時 (段階 2) においては Sequential Read (Q= 32,T= 1) が高速であることが, 端末の起動時間短縮に貢献したと考えられる.

この転送性能は端末内蔵の NIC の性能を大きく上回るものであるため, 端末を 1 台のみで起動する場合においても, 単純なネットブートよりも適切にキャッシュを活用する方が好ましいことがわかる.

4.4 ディスクイメージ更新により作成される差分ディスクの分析

ディスクイメージの更新の影響を調査するために, 手動更新によりアプリケーションを 1 つインストール*3して,

*2 段階 1a: TFTP によるブートローダの転送
 段階 1b: ブートローダによるカーネル等の読み込み
 段階 2: カーネルによるサービスやアプリケーションの読み込み
 *3 秀丸エディタをインストールした

生じる差分ディスクと更新直後の端末起動の挙動とを分析した。

この際にサーバー側の vhd ファイルに生じた差分はファイルベースで 5.28 GB であったが、コンテンツベースではわずかに 340MB であった。このとき作成されたディスクを端末に割り付けて起動したところ、生じたトラフィックは 142.6 MB であった。

導入したのはアプリケーションであり、端末起動時には読み込まれることはない領域のため、本来は端末起動時にトラフィックは生じないと想定していた。そこで、ここで端末起動時に転送しているデータを分析したところ、大半はファイルシステムのメタデータとシステム情報（レジストリ）であった。これらのデータは更新作業に直接は関係していないにも関わらず転送を要しているのは、Windows がこれらの重要な領域を定期的にバックアップしつつ、起動のたびにローテーションしていることに起因すると考えられる。

想定より多くの転送をしていることが確認されたが、評価環境において 1 教室の 70 台の端末を一斉に起動して各端末が 150MB の転送を行ったとしても、その転送は約 40 秒で完了することになる。この転送に要する時間の分だけ端末の起動は遅延することになるが、それほど大きな遅延ではないことと、この遅延が生じるのは更新作業を行った直後の 1 回限りであることから、運用への影響は大きくないと考えられる。

4.5 自動更新により作成されるディスクの分析

自動更新が繰り返し実施されているが、そこで作成されている差分ディスクイメージのサイズは平均するとファイルサイズベースで 7.4GB と大きく、コンテンツベースのサイズでも 1.12GB であった。

自動更新の場合には更新状態にした端末を長時間放置するため、手動更新の場合と比してサイズが大きくなったものと思われる。キャッシュ全体をキャッシュするソリューションを利用するとディスクイメージを自動更新するごとに少なくともこのサイズの差分ディスクを各端末に転送しなければならないことになり、毎週のように自動更新を行う環境での利用は実用的とはいえない。

なお、比較のために自動更新を行う前のディスクで端末を起動してキャッシュを貯めた後に、更新後のディスクで端末を起動した際のトラフィックを計測したが、手動更新の場合と同程度であった。自動更新では、Windows Update や ウィルスパターン更新が行われたものと思われるが、端末起動時にすぐに必要となるモジュールが更新される頻度は高くないことによるものと思われる。

5. まとめ

端末教室で利用する管理ソリューションとして 4 つの

管理方式を比較し、ネットブートを基盤としつつ部分的にキャッシュを持つ方式が最適であることを実機検証により確認した。キャッシュが有効であればサーバーとの通信はほとんど発生せず、ディスクイメージ更新直後においても転送される量は 150MB 程度であることを確認した。また、ネットブートサーバーをデータセンターに設置した構成で評価を行ったが、1 教室程度の端末の一斉起動では運用に影響をおよぼすような起動時間の遅延は見られなかった。

一方、ディスクイメージの更新作業においては、書き込み処理が生じる都度サーバーとの通信を要するため通信遅延の影響を大きく受ける。今回評価を行った環境ではネットブートサーバーと端末との間の RTT は平均 0.3ms 程度と良好な部類であったが、それでもアプリケーションの導入などにおいては通常よりも時間がかかるように感じられた。今後ネットブートサーバーをデータセンターに設置する構成を採用する上では、通信遅延の評価や更新作業を実施する際の挙動に対する対策が求められる。

次に、サーバー上でのディスクイメージの管理においては、標準的なフォーマットである vhd や vhdx 形式が扱いやすいが、ファイル容量という観点では無駄が多いことを確認したが、標準的ではないフォーマットを用いると運用コストの増大が危惧されるため、ファイルフォーマットは vhd, vhdx 形式を利用しつつ、サーバー側のファイルシステムの圧縮機能を有効にしたり、ストレージの Dedup 機能をの活用する対策が考えられる。

さらに、多数の端末をイメージ更新直後に一斉起動した際の起動遅延は、端末教室の上流ネットワークの輻輳が原因であることを確認した。この輻輳を回避し起動時間を短縮するために、転送するデータを圧縮したりイメージ更新時にメタデータをローテーションしないようにする手法が考えられる。

参考文献

- [1] 八田 直樹, 丸山 伸, 松川 正義, 西村 浩二, 相原玲二, “ネットブート環境における読み込みキャッシュ機構の改善による起動時間短縮の試み”, 情報処理学会, 研究報告インターネットと運用技術 (IOT), 2012-IOT-16 (14), 1-6 (2012-03-08)
- [2] 丸山 伸, “ネットブート型 Windows 端末システムにおける起動時間短縮への試み”, 情報処理学会, インターネットと運用技術シンポジウム 2015 論文集, 2015, 57-57 (2015-11-26)
- [3] “[MS-VHDX]: Virtual Hard Disk v2 (VHDX) File Format” available from <https://msdn.microsoft.com/en-us/library/mt740058.aspx>

謝辞 本研究における評価作業においては、徳島大学技術支援部 八木香奈枝氏, 板東孝文氏に特にお世話になりました。この場を借りて御礼申し上げます。