

Title	JAIST統合ユーザ環境の改良
Author(s)	宮下, 夏苗
Citation	国立大学法人北陸先端科学技術大学院大学技術サービス部業務報告集 : 平成23年度: 41-48
Issue Date	2012-08
Type	Others
Text version	publisher
URL	<a href="http://hdl.handle.net/10119/10803">http://hdl.handle.net/10119/10803</a>
Rights	
Description	

# JAIST 統合ユーザ環境の改良

宮下 夏苗

情報社会基盤研究センター

## 概要

当情報社会基盤研究センターは2010年2月より、それ以前にサービスしていたWindowsターミナルサーバシステムに大幅な改良を加え、ターミナルサーバクラウド環境として全学的サービスを開始、運用を続けていることを、2011年の業務報告集に述べた。本稿ではJAISTにおいてこのターミナルサーバクラウド環境を含む統合常用情報インタフェースの改良の履歴と、現在の運用および改修について述べる。

## 1 統合常用情報インタフェース

情報社会基盤研究センターは従来、組織内の教職員、学生の日常的利用に充てる情報インタフェースとして、“いつでも”、“どこからでも”利用できる、個人用統合デスクトップ環境を構築、提供してきた。本環境は優秀な個人用の文房具であると共に、本学の有するあらゆる情報資産にシームレスにアクセスできること、個人が日常の情報インタフェースとして利用するに十分なスペックを有することが求められる。

本学で2010年2月に公開したターミナルサーバクラウド環境の仕様を検討するにあたって、サーバホステッドVDI方式を採用したターミナルサーバ型のデスクトップサービスを選択した。リモートデスクトップサービス(RDS)によるターミナルサーバ型のデスクトップサービスを既に全学運用しており、運用実績があったこと、また、システム導入当初ユーザに配布していたクライアント端末機は、ローカルリソースが若干非力であったことも主な理由と言える。

ここでは、情報インタフェースを構成するシステムを運用するにあたり必須となる、システムのメンテナンス性に焦点を絞り、環境の変遷に伴うこれらの運用方式の工夫と改良、今年一年間で行った運用改善について順に述べる。

## 2 メンテナンス

ユーザに提供する情報インタフェースを構成するシステム自身の運用にあたり、障害対応、調査はもとより適切な日常のメンテナンス、すなわち修正パッチの適用やバージョンアップは必要である。物理機はもとより、現時点で我々が運用している仮想環境ではさらに多段に、ハイパーバイザのレイヤ、VDI環境の制御、管理を司るレイヤにおいてもバグフィックス、アップデートは随時調査し、必要に応じて動作検証・適用することは管理者の必須業務である。

だがある意味でそれ以上にユーザの意識に上るのが、ユーザに提供しているインタフェースの更新である。デスクトップOS自身のアップデート版、または修正用のパッチが公開されることもあれば、前項にて述べた障害や不具合の対応、あるいは利便性の向上のため、サービスの公開後に設定ファイルやカーネルパラメータ、レジストリ等改修が必要になることもある。また、そこでサービスするアプリケーション群が多ければ多いほど、アプリケーションごとにアップデート版のリリースがあり、問題があればその修正が必要となってくる。

更新が行われないデスクトップ環境は“古く”なる。既知のバグフィックス、脆弱性対応のセキュリティパッチが適用されていなければ、障害が発生した場合にそのデスクトップを利用するユーザに直接影響を与え、悪くすれば個体の仮想マシンのみならず広範囲に影響を及ぼすこととなる。

障害の有無に関わらず、デスクトップ環境が“古い”ということは、それを利用するユーザがすぐに認識す

る問題となる。

新しいコンテンツに対応できないなど実利的な不具合をもたらすことはもちろん、古いバージョンを利用していること自体がユーザに違和感やストレスを与える要因となることも大いに考えられる。

このため、ユーザに提供すべきデスクトップ環境のインタフェース更新は、サービス提供者側の重要な課題となる。

まず、システムの更新、とくにユーザインタフェースの更新が管理者側からみてどのような作業となるかを考える。

パッチ当てや修正作業は、既存のインストールに対して修正を適用し、構成するライブラリやアプリケーションを上書きする作業となる。バージョンアップなど、対象がアプリケーションであればその時点のインストールに影響を与えず、追加でのインストールが可能な場合もあるが、特に Windows 系 OS のアプリケーションはバージョンが上がった際に前回のインストール情報を引き出し、これを更新するような動作になるのが一般的である。古いインストールを残しておくメリットより、適切に修正された更新版を利用するメリットのほうが一般的に高く、またアプリケーション自体複数バージョンの同時利用を想定しておらず、不具合が起きる可能性もある。

また、更新の対象がユーザによって利用されている場合は、その対象を上書き、更新することはできない。デスクトップ環境を構成する OS 自身はもとより、そこで利用されているアプリケーションやドライバ、ライブラリについても同じことが言える。

これは、更新中はそのシステムを利用できないことを意味する。OS 自身であれば、基本的にその OS から何らかの方法でユーザをログアウトさせる必要があり、アプリケーションであれば、そのアプリケーションの利用を停止しなくてはならない。

もし作業中に代替となるシステムがなければ、ユーザが利用再開までその OS、アプリケーションの更新終了を待たなくてはならない状況も発生する。

これを踏まえて、重要となる点が 2 つある。

- どのように、対象をユーザの利用から切り離すか
- 複数台の対象機に、どのようにアップデートを適用するか

ユーザに適切な環境を提供することを目的とした上記 2 つの管理上の観点から、これまで改修を重ねてきた統合的常用情報インタフェースの歴史と、今年度行った改良について述べる。

### 3 ~2006

#### 3.1 Sun Workstation

2006 年に全学向けのリモートデスクトッププロトコル (RDP) を利用したターミナルサーバ型の Windows デスクトップサービス提供を開始する以前にも、Sun Solaris を利用し、全学のユーザが個々の日常の文房具として利用するために、『いつでも』『どこからでも』同じように使い、計算サーバからストレージまであらゆるリソースにシームレスにアクセスできる、高度に統合化されたデスクトップ環境を設計、開発し、用途に応じた 2 形態で提供していた。

##### 3.1.1 fat client 型

自分自身でインストールやシステムの改変を行いたいユーザ向けに、SolarisOS のデスクトップ型ワークステーションを一人一台の割合で配布した。ワークステーションはファイルサーバ上にあるログインユーザのホームディレクトリをオートマウントするよう設計され、ユーザは「どのワークステーションを利用しても」自分用のワークステーションにリモートログインすることができるほか、他のワークステーションからホームディレクトリに格納された設定情報を元にした個人のデスクトップ環境を利用することもできた。また、ワークステーション本体に個人的にインストールしたツールを利用することもできた。

### 3.1.2 thin client 型

自分自身でメンテナンスまでは行わない、基本的に提供されている環境、ツールが利用できれば良いユーザ向けに、SolarisOS の共用サーバを配置していた。ユーザが手元の端末からこれらの共用サーバにアクセスした際にファイルサーバ上にあるホームディレクトリがオートマウントされ、ここにある設定情報を利用することで、他の共用サーバを利用しても同じデスクトップ環境を利用することができた。次項に述べるターミナルサーバ環境と、OS は異なるがその構成は似通っている。

## 3.2 メンテナンス

これらのシステムについて、提供するインタフェースを更新するための構成、設計および実際の更新について述べる。

### 3.2.1 対象の切り離し

ユーザが利用するアプリケーション、ツール群を NFS 共有ディレクトリに置き、OS が自動的にディレクトリをマウントする構成としていた。ユーザのデフォルトパスにもこのディレクトリが加えられており、全ユーザはディレクトリ内の更新に応じて新しいツールを利用できた。ディレクトリの更新はセンター職員のほかシステムに精通したユーザグループがこれを担当することで、ユーザのニーズに合った更新を展開できていた。対して、サービスが個々のシステムの提供であり、負荷分散が行われていたわけではないことから、システム自身のアップデートにはユーザアナウンスとユーザの協力を必要とした。

### 3.2.2 一括アップデート

前述の形式により、アプリケーションについては共有ディレクトリの利用とユーザグループの協力により、必要な更新をコンスタントに適用できた。また OS 自身の更新も一括処理が可能なように工夫されており、全クライアントがシステム起動時にネットワークボリュームを読み込み、必要なアップデートがあれば読み込んでから起動する設計となっていた。また、クライアントの障害に対応できるようブートイメージをネットワークに置き、起動時のコマンドオプションで随時ネットワークからの上書きインストールが可能なように設計していた。

しかし、クライアント OS にネットワーク経由でアップデートを行った回数は比較的少なく、1 機種について年に 1 度あるかないかであったと思われる。ブートイメージの作成手順と検証にそれなりの労力がかかること、実際にアップデートするにはネットワークボリューム上に適切なイメージを設置したうえでユーザアナウンスにより端末の再起動を依頼し、ブートイメージの読み込みを行う手間を要したことが理由として挙げられる。また、主要なアプリケーションが NFS 共有ディレクトリ上に置かれてシステムと完全に分割されており、システム自身のアップデートへの要求が強くなかったこともある。

## 4 2006 ~2010

### 4.1 Windows Terminal Server

上記 Solaris 環境の根本的な問題として、当時増えつつあった Microsoft Windows 系 OS を動作条件とする研究、教育用ソフトウェアのサポートが困難であった。

このため 2006 年、Windows Terminal Server システムを導入し、Windows 系 OS によるデスクトップ環境の提供を開始した。各研究室ごとに Windows 2003 Server が動作する物理サーバをターミナルサーバとして配備し、ファイルサーバに格納されたホームディレクトリ、共有ディレクトリをこれらターミナルサーバからマウントする設計を追加した。また旧 Solaris 環境の利用も維持するべく共用の Solaris サーバを準備し、ユーザがターミナルサーバ、および Solaris サーバに接続するための端末を配置した。

### 4.2 メンテナンス

ユーザに提供されるデスクトップ環境として、この時期のターミナルサーバは、個々のシステムのメンテナンスが非常に困難であった。このシステムについて、提供するインタフェースを更新するための構成、設

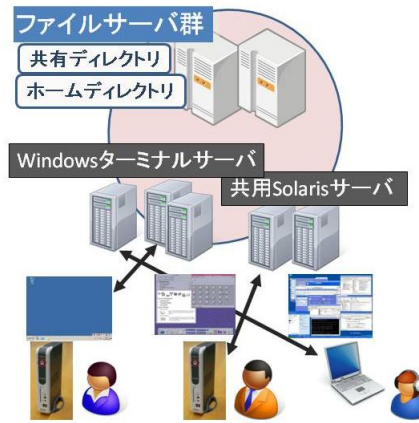


図1 2006年 ~2010年

計および実際のシステム更新状況について述べる。

#### 4.2.1 対象の切り離し

システム上、これまで利用していた SolarisOS のようにアプリケーションを共有ディレクトリに置くことが必ずしもサポートされておらず、通常のアプリケーションの多くが規定のドライブ直下に置かれることを想定して作られていたために、アプリケーションが他のドライブに置かれることで正常に動作しない場合もあり、以前の方式を踏襲することは困難だった。また、前システムと同様ユーザが個々のサーバに直接ログインする方式を取っており、ユーザにサービスを公開するうえでログインセッションを管理する方法はこの時点では実装されていなかった。このためアップデート等でシステムをユーザの利用から切り離す場合には必ずアナウンスに基づくサービス停止が必要だった。

#### 4.2.2 一括アップデート

各サーバを研究室ごとに配備し、アップデートやインストールを各研究室の管理者に委ねる方法を採用したことで、OS の更新、アプリケーションの更新に伴う実質のサービス停止タイミングをユーザ都合になるべく添わせることとしていた。これは反面、一括更新、中央集中管理とは対極にある状況であったと言える。管理者が適切にサーバのアップデート、維持管理を行える状況であれば良いが、そもそも管理者自身が Windows サーバの扱いに慣れたユーザばかりではないこと、また管理者の卒業、転任などによる不在や情報伝達の不足などもあり、適切なアップデートが行われぬままシステムが使い続けられる状況も発生していた。

### 5 2010 ~2011

#### 5.1 Windows Terminal Server Cloud

上記システムの問題点を改良すべく学内プライベートクラウド環境を構成したことは昨年の業務報告集にて述べた。

#### 5.2 メンテナンス

##### 5.2.1 対象の切り離し

サーバ仮想化により、動作中の仮想ホストを無停止で異なる物理ホストに移動する機能を活用して物理ホストのメンテナンス性の確保し、物理リソースの負荷分散を実現した。また、ユーザセッションの開始要求をゲートウェイに一元化し、接続先ターミナルサーバに割り振る方式を採用し、ユーザセッションの負荷分散を実現した。これに伴いターミナルサーバを負荷分散の対象から解除することで、ユーザサービスに影響なくサーバを利用対象から切り離すことができた。

## 5.2.2 一括アップデート

クラウドシステムを構成し、ユーザインタフェースを提供する仮想マシンは、2010年の時点では総計100台を超えるWindows機であった。これらのホストに対して、一台ずつセキュリティパッチを適用する、アップデートを適用するといった作業が非常に困難であったため、これらの作業を緩和、軽減する方式として、以下の方式を採った。

1. WSUS(Windows Server Update Service)
2. リモートスクリプト
3. App-V

**WSUS** WSUSはMicrosoftの機構で、Windows Updateを管理サーバで集中管理し、管理クライアントのパッチ適用状況を確認する、必要なパッチを選択適用させるといった一括処理を可能とするシステムである。2006年時のターミナルサーバシステムから導入しており、運用実績のある本サービスをこのクラウドサービスに適應するよう構成し、運用している。

Microsoftが提供する一般のアップデートリリースについては、この機構によって全仮想マシンのアップデート適用を制御する。

**リモートタスク** WSUSで配布する以外のアップデート適用、その他の仮想マシンへの一括操作のために、リモートスクリプトを作成、利用している。コントロールサーバ上に仮想マシンへのタスクスケジュール送信用バッチファイルを準備し、各仮想マシンがインストーラ実行用バッチファイルを実行するよう、リモートからタスクスケジュールコマンドを投げる方式を採った。

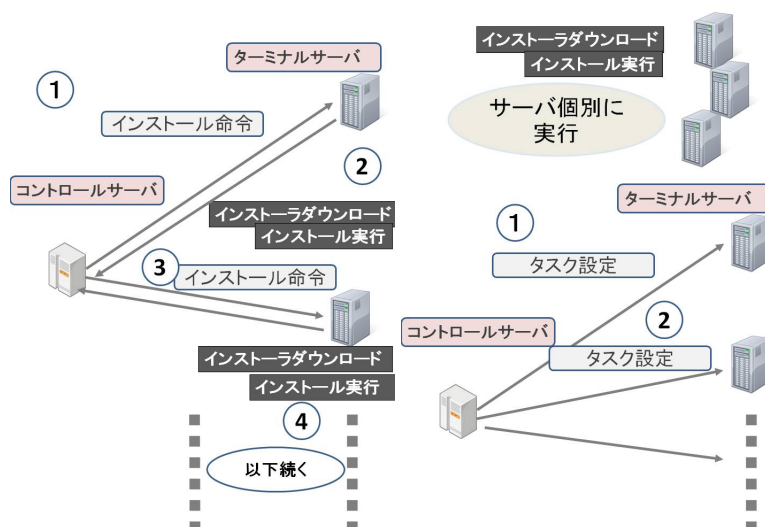


図2 アップデート方式改良前後

**アプリケーション仮想化** Microsoft Application Virtualizationに基づいたアプリケーション配信システムにより、アプリケーションをパッケージングし、全サーバに配信するシステムを利用した。全体へのインストーラ適用、アップデートといった作業を劇的に簡略化することができる。また、同じアプリケーションの異なるバージョンを同時に配信する、所有ライセンス数に応じて、同時に利用しているプロセス数の監視・管理を行うといった機能も活用している。

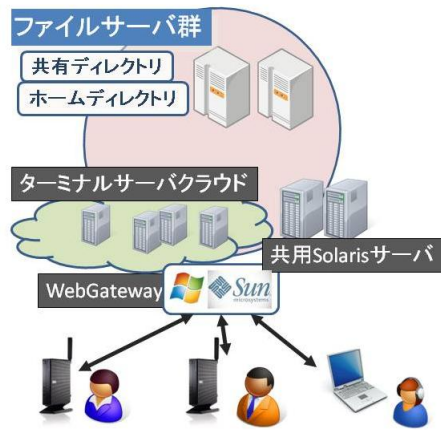


図3 2010年～

表1 システムメンテナンス作業環境の改善

年代	OS	メンテナンス時の作業対象切り離し	一括アップデート
～2006	Solaris	アプリケーションは随時可能 システムメンテナンスは要アナウンス	可能
2006～2010	Windows 2003	機器ごとに個別調整	困難
2010～2012	Windows 2008	システム、アプリケーションともに 利用対象から切り離し可能	可能

## 6 2011～2012

ここまで構築されたシステムで、2006年以降のWindowsターミナルサーバシステムのメンテナンス環境に関する問題点は大きく改良された。

しかし、まだいくつかの問題が残っている。

### 6.1 問題

システムを負荷分散の対象から外すことで利用するユーザがいない状況を作り出し、この間にアップデートを行う方式を可能にしたことは、ユーザビリティの面でもメンテナンス性の面からも、これまでには無かった改善であった。だがこのことは、アップデート中はサービスを提供するサーバが台数的に減ることを意味する。翻って、ユーザの利用は管理者によるアップデートを考慮したものではないため、サーバの負荷が減るわけではない。ユーザに十分なリソースを提供するにはできる限りダウンタイムを減らし、速やかにアップデートを完了してユーザの利用対象に再度組み込むことが必要となる。

しかし、ごく一般的な管理者の作業の優先順位としては、通常はアップデートより障害対応が先に来る。目の前で起きている障害を前に、メーラーのマイナーバージョンアップなど行っていない。

また当センターはユーザからの問い合わせも受け付けており、ユーザからの問い合わせに対する対応は、業務時間内においては各自のできる限り優先的に取り扱うこともなっている。

つねにアップデートを最優先で行うことはできず、実情はサーバを利用対象から外したまま一週間近くアップデートを完了させられないこともあった。

また別な問題として、あくまでインストール、アップデートはシステムに対し新たなパッチ、アプリケーションを利用可能にするものであり、たとえばユーザ個々の初期設定(個人プロファイル)部分に改修を要する場合には対応が困難であった。

この問題を回避するため、アップデートに新たな方法を取り入れた。

表2 4-6月期の研究系 TS 障害対応アップデート事例

4月	OS 更新に基づくプリンタドライバ一括更新	リモートタスク
4月	Outlook の複数起動によりレジストリが一部破損する	リモートタスク
4月	Citrix バグによるログイン障害	リモートタスク
4月	Adobe Reader アップデートによるユーザ権限での再起動障害	リモートタスク
5月	MMS のリンクが動作しない	リモートタスク
5月	APP-V のキャッシュサイズによるローカルディスク容量不足	リモートタスク
6月	システムの予期しない再起動障害	リモートタスク
6月	Word の複数バージョン起動によるエラー	ログオンスクリプト, リモートタスク

表3 4-6月期の研究系 TS ソフトウェア更新事例

4月	Adobe Flash Player	GPO インストール
4月	Java	GPO インストール
5月	Mozilla FireFox, Thunderbird 12.0	リモートタスク
6月	Mozilla FireFox, Thunderbird 13.0	リモートタスク

## 6.2 一括アップデート

GPO インストール アップデートの適用に、Windows の GPO(グループポリシー) によるソフトウェアインストール機能をテストし、採用した。適用すべきインストーラを準備し、設定しておけば、システム再起動時に自動的にソフトウェアがインストールされる。

この機能を利用することで、システムを負荷分散の対象から解除することなく、定期再起動とともにアップデートを適用することが可能になった。

ただし、この方法は適用できるインストーラが msi 形式のものに限定されており、exe、msp などにより提供されるインストールについてはまた別な方法を取る必要がある。

スタートアップスクリプト アップデートの適用、一括設定変更等に、Windows の GPO(グループポリシー) を利用したスタートアップスクリプト、ログオンスクリプトをテストし、採用した。システムの起動時、あるいはユーザのログオン時にバッチスクリプト、あるいは powershell、VBS スクリプト等を動作させることができる。

ログオン処理、再起動処理のたびに動作させられるため、ユーザの初期設定にあたるレジストリ、個人プロファイル部分に変更が必要な場合にも適用が可能となった。

## 6.3 実際のアップデート

2012年4月1日、研究系ターミナルサーバの OS を、これまでの Windows Server 2008 から、Windows Server 2008 R2 へと更新した。新システムであることから、更新以降複数の不具合修正、システム改良が必要となった。ここまでに述べたシステム更新方法を用い、研究系ターミナルサーバのみで3ヶ月間に述べ8回の不具合改修および修正パッチ適用、4回のローカルアプリケーションのバージョンアップを行ってきた。

一般的に、不具合を発見してから適切な修正方法を検討、調査するために多くの時間と労力を要する。多くの場合、その間ユーザに不具合のあるシステムを提供し続けなくてはならないため、修正方法が確定したあとは可能な限り速やかに、かつ確実に更新を反映できるよう、これらのアップデート方法を用いて逐次更新を行っている。このような不具合の調査と修正、システムのアップデートと改修を現在の主な業務の一部として行っており、この項ではこれらの業務を効率化するために利用を試み、実用に充てている手法の一例を挙げた。



## 7 まとめ

本稿では JAIST における歴代のユーザ環境と、その改良、これらに基づいて構築された現在のターミナルサーバクラウドシステムにおけるメンテナンスについて述べた。

歴代のシステムを通覧し、Solaris をベースに 2006 年以前に構築されていたシステムは非常に機能的かつメンテナンス性に優れた中央管理型のシステムであったといえる。利用アプリケーションの対応という根本的な問題により Windows をベースにしたシステムへと更新したが、対応できるアプリケーションが大きく増えたことと引き換えに、一旦はその利点が失われてしまった。

過去のシステムの利点と欠点を踏まえ、その設計思想を理解して、現在のシステムをより機能的でユーザビリティの高いシステムへと更新するのが現在の一業務である。