

遊休 PC を用いた仮想デスクトップサービスの構築と課題

和田智仁¹⁾

1) 鹿屋体育大学 スポーツ情報センター

wada@nifs-k.ac.jp

Remote Desktop Service with Computer Lab PCs

Tomohito WADA¹⁾

1) Information Technology Center for Sports Sciences, National Institute of Fitness and Sports

概要

2020 年現在、新型コロナウイルス感染症対策のため大学の教室などでは収容定員を減らした運用が行われている。鹿屋体育大学の PC 教室においても使用台数を減じた運用を行っており、教室に設置された PC のうちおよそ半数が利用できない状態となっている。そこで、これらの遊休 PC を使った仮想デスクトップサービスについてプロトタイプシステムを構築し、その実現可能性を検討した。

サービス試行を目的に、リモートデスクトップ接続先を管理する Web アプリケーションを開発し、遊休 PC には必要な設定を行った。サーバシステムは Node.js の Web アプリケーション・フレームワークである Express を用いて開発した。試行の結果から技術的にはサービスの実現が可能であることを確認できたものの、運用に関わる課題も明らかとなった。

1 はじめに

2020 年に入り新型コロナウイルス感染症が日本国内においても急速に広まった。文部科学省は 3 月 24 日付けの通知^{*1}において「大学等では、日常において、3 つの条件が重なることを徹底的に回避する対策が不可欠である」とし、学生に関する配慮として「教室等の利用に当たっては、短時間の利用としたり、一斉に使用しない」などの対策を求めた。

このような中、鹿屋体育大学においても 2020 年 4 月には教室の収容定員を半数程度に減らすなどの対策が行われることとなった。PC 教室についても他大学の事例 [1] などを参考に、使用可能な機器を半数程度に間引いた上で運用を行うこととした。しかし、使用可能台数の減少により例年どおりの方法で PC 教室での授業を行うことは不可能となった。これにより筆者が担当する情報関連科目「情報処理 A」も、収容人数が大きな一般教室が割り当てられ、対面授業を実施する際には一般教室を使用することになった。

鹿屋体育大学では 2015 年よりタブレット (Apple iPad) の必携化を行っている [2] ことから、一般教室で

もタブレットを使用した一斉演習を行うことは可能である。そこで当初はタブレットを使い例年通りの演習を行う計画を立てた。しかし、タブレットと PC とでは操作方法などの相違点も多く、特に Office 系のアプリケーションにおいてタブレットアプリでは機能が不足するなど、限界が感じられた。

これらの授業が行われる時間帯は、PC 教室は授業のために予約されており、全 PC が遊休状態となる。また、感染症対策のため、およそ半数の PC は常時遊休状態となっている。そこで、これらの遊休 PC を使い、リモートデスクトップを個別にホストする形式で仮想デスクトップサービスを実現できないか検討することとした。本稿では、構築したプロトタイプシステムの概要と、それらを用いたサービスの課題について報告する。

2 システム設計

今回のプロトタイプ実装では、簡易的に 1 台の管理サーバを設置し、各機能を実現することとした。図 1 にシステムの構成と利用手順を示す。

利用者はブラウザを使って管理サーバにアクセスする (手順 1)。管理サーバは管理情報から利用可能な PC を検索し、空きがある場合には利用条件を提示する。利用者がブラウザに表示された条件を承諾すると

^{*1} 令和 2 年度における大学等の授業の開始等について (通知) https://www.mext.go.jp/content/20200324-mxt_kouhou01-000004520_4.pdf

(手順2), 管理サーバは Wake On Lan (WOL) を発出して PC を起動し (手順3), 起動した PC の情報を利用者に伝える (手順4). 起動された PC ではリモートデスクトップサービスが有効化されており, 利用者は管理サーバから得た PC 情報をもとにリモートデスクトップソフトウェアを使用してリモートデスクトップ接続する (手順5). PC では利用者のログオン・ログオフ時に, 自身の利用状況を管理サーバに通知し (手順6), 管理サーバは PC の利用状況としてその情報を管理する.

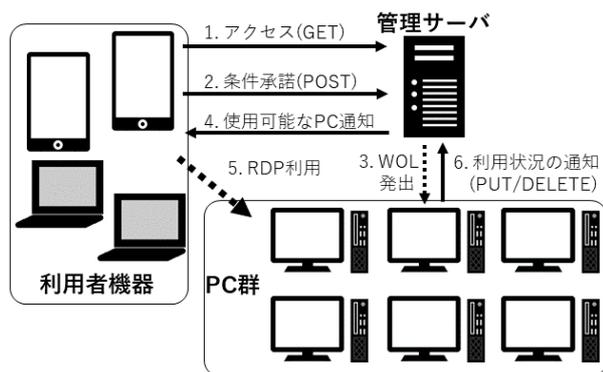


図1 システム構成および利用手順

なお, 今回の実装では REST (Representational State Transfer)[3] と呼ばれるアーキテクチャスタイルを採用することとした. REST では全ての情報を URI で表現し, HTTP メソッドを用いて情報を操作する. 図1 のかっこ内表記は使用する HTTP メソッドを表している.

3 PC 群の設定

PC 教室においては, 従来より機器管理のため WOL 機能は有効化されていたため, 試行に伴う設定変更は不要であった.

各 PC のリモートデスクトップサービスは, Active Directory のグループポリシーを用いて一斉に有効化した. 具体的には, グループポリシーの管理用テンプレートから, Windows コンポーネント/リモートデスクトップサービス/リモートデスクトップセッションホストにおいて以下の設定を行なっている.

- 【接続】ユーザーがリモートデスクトップサービスを使ってリモート接続することを許可する：有効
- 【セッションの制限時間】 アイドルセッションの制限：15 分, 切断されたセッションを終了する：15 分
- 【リモートセッション環境】 [シャットダウンダイ

アログ] から [切断] オプションを削除する：有効

ここでは 15 分のアイドルまたは切断を検知すると自動的にセッションをログアウトするようにした. これはアプリの強制終了などで, 接続状態を保ったままサービス利用を終了した場合の対策である.

手順6 の PC から管理サーバへの利用状況の通知については, ログオン/ログオフスクリプトを用いた. これらのスクリプトも Active Directory グループポリシーを用いて一斉に展開した. スクリプトでは OS に標準で用意されている curl コマンドを用い, 環境変数であるコンピュータ名とユーザ名を管理サーバに送出する. ログオンスクリプトの内容を以下に示す. 送出時, ログオンの場合は HTTP の PUT メソッド, ログオフの場合は DELETE メソッドを用いている. Web と REST の採用により管理サーバとの通信については非常に簡便に実現することができた.

```
curl --request PUT ^
http://(管理サーバ)/%COMPUTERNAME%/USERNAME%
```

4 管理サーバの構築

管理サーバは, Node.js^{*2}および Web フレームワーク Express^{*3}を用いて実装した. Node.js は“非同期型イベント駆動のサーバサイド JavaScript 環境”であり, Express とともに用いることで Web アプリケーションを簡便に構築できる. また, Node.js ではパッケージ管理ツール npm を用いて各種の機能を追加できる点にも特徴があり, 今回は WOL マジックパケットの発出のために wake_on_lan パッケージを利用した.

今回の試行では, 既存の仮想環境上に CentOS 8.1 サーバを設定し利用した. WOL を発出するため, サーバは PC 教室と同一の VLAN に接続した.

プロトタイプでは管理する PC 台数が少ない (85 台) ことからデータベース等を用いず, PC の情報 (マシン名, IP アドレス, mac アドレス) を JSON ファイルに保存しておき, サービス起動時にそれを読み込み, PC の利用状況とともにメモリ上で管理することとした.

*2 <https://nodejs.org/ja/>

*3 <https://expressjs.com/ja/>

5 利用者側画面の設計

サービスの利用に際し、利用者は管理サーバに対してリモート接続可能な PC の問い合わせ (GET メソッド) を実施する。具体的にはブラウザから管理サーバへのアクセスを行うことになる。本システムではこの際に利用条件を利用者に提示し (図 2)、条件に同意した後接続可能な PC 情報を提示することとした。



図 2 利用開始時の Web 画面

この HTML ドキュメントには jQuery の Ajax を用いたスクリプトが設置されており、同意ボタンの押下により管理サーバに対する POST メソッドでの http アクセスが行われる。

管理サーバは POST でのアクセスを受けると、空き PC を WOL で起動し、起動した PC 名をクライアントに回答する。この回答を受けると図 3 のように使用可能な PC 名が利用者の Web 画面に表示される。起動から接続可能となるまでには 60 秒程度を要することから、PC 名を提示する画面においては、JavaScript を用いたカウントダウンタイマーを表示している。

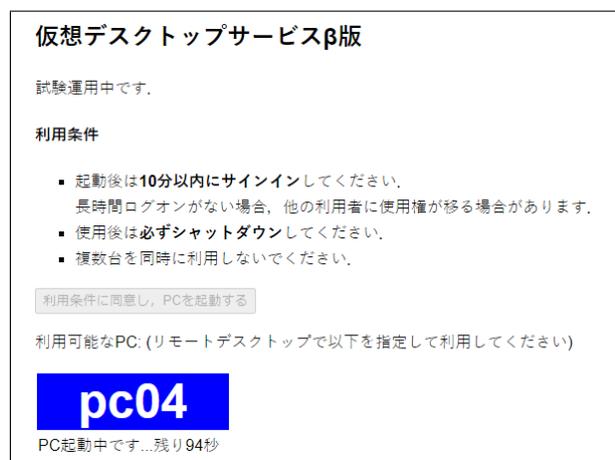


図 3 同意後の Web 画面

6 考察

今回のプロトタイプシステムでは、Node.js と Express を用いたことで Web アプリケーションの開発を簡便に行うことができた。また、REST の採用で API を簡素化したことから、PC と管理サーバ間の通信も OS の標準的な機能のみを使って簡便に実現できた。試行を行っている期間のサーバ動作も安定しており、この構成のままでのサービス開始も十分可能であると考えられた。

ただし、本システムを構築し試行を行う中で、いくつかの問題点も考えられた。

一つ目は、管理サーバの機能として、一人の利用者が同時に複数台を利用することを防ぐ仕組みがない点である。ただし、この問題は物理的な PC を利用する際にも起こり得ることであり、この対策を講じるのであれば仮想サービス側ではなく物理的利用を含めた形での対策を行うことが適切であると考えられる。

二つ目は、意図的かどうかに関わらず、過去に接続した PC 情報からリモートデスクトップ接続する利用者が予想される点である。利用者が使用するリモートデスクトップアプリでは、過去の接続履歴が保存され再度接続先を入力する手間が省けるようになっている。この情報を使って安易に PC 接続を試みる利用者がいた場合、正規の利用者が迷惑を受けることになる。この解決策としては、一時的に有効な DNS 名を作成し、管理サーバがそれを利用者に提示する、といったことが考えられる。ただし、この方法では DNS と連動したやや大掛かりなシステム開発が必要である。また、IP アドレスを使って直接接続するような行為があった場合にはそれを防ぐことはできない。

三つ目は、PC のシャットダウンについてである。現時点でシャットダウンは利用者の操作に依存しており、有効な PC 電源の管理方法を見つけられていない。電源を入れたままの PC が長時間放置されないようにするために、何らかの対策が必要であると考えている*4。

これらの点は、ユーザの利用方法が適切であれば問題にはならない。そのため本学程度の規模では実質的には問題とならない可能性もあるが、技術的な検討は継続したいと考えている。

*4 現時点では、既存の PC 管理システムにスケジュール機能があるた、夜中に一斉停止を行うことは可能であった。ただし、24 時間運用のサービスとした場合にこの方法は利用しにくい。

また、本稿で検討しているサービスのように、教室に設置された PC をリモートから利用する形態については、ソフトウェアの使用許諾条件に注意する必要がある。本システムでは純粋な仮想化は行っておらず、かつ同時に 1 人が 1 台の PC を利用する形態ではあるが、このような形態でもリモートでの利用に制限を設けられている場合がある。例えば、Microsoft 社の製品条項 [4] のデスクトップオペレーティングシステムの項には、「3. ローカル使用は、すべてのユーザーに対して許可されますが、4. リモート使用は、ライセンスを取得したデバイスの主要ユーザー、および別のライセンスを取得したデバイスまたは Windows VDA のライセンスを取得したデバイスを使用するその他のユーザーに対して許可されます。」との記載が見られる。本試行を通じて、大学で所有するライセンスとそれに付随する各種特典の内容を踏まえつつ、接続要件等を整理しようと試みているものの、その作業は困難を極めている。一般に利用者の持ち込み機器の場合には別途ライセンスが必要になる場合も多く、三島らも仮想デスクトップサービスにおいて「特有のライセンスコストが発生する」と指摘している [5]。現時点では、本稿で提案するような形態でのサービス提供は、ソフトウェアライセンスの面で難しくなると予想している。

7 まとめ

本稿では、新型コロナウイルス感染症対策として PC 教室で利用できなくなった遊休 PC を用いて、リモート接続を可能とする仮想デスクトップサービスについて検討を行った。試行するためのシステムを構築

し、基本的なサービスの実現は簡便に行うことができることを確認した。実際の運用に向けては、いくつかの課題点が挙げられた。また、PC のリモート利用に関してはソフトウェア使用許諾条件の確認が必要で、PC 教室のように多数の利用者が同一 PC を利用したり、利用者が所有する個人デバイスでの接続したりする場合には許可されないケースがあるため注意が必要である。

参考文献

- [1] 松浦健二, 金西計英, 遠隔×対面での授業実践におけるライブ授業実施および支援組織の活動に関する一例, 教育システム情報学会誌, Vol. 37, No. 4 [草稿版], 2020.
- [2] 和田智仁, 高橋仁大, 中村勇, 体育系単科大学におけるタブレット必携化, 第 40 回教育システム情報学会全国大会予稿集, pp. 119–120, 2015.
- [3] Roy T. Fielding, Richard N. Taylor, Principled Design of the Modern Web Architecture, In proceedings of 22nd International Conference on Software Engineering (ICSE' 2000), Limerick, Ireland, June. 2000.
- [4] 製品ライセンス条件- マイクロソフト ポリウムライセンス, <https://www.microsoft.com/ja-jp/licensing/product-licensing/products> (2020/10/16 アクセス).
- [5] 三島和宏, 櫻田武嗣, 鈴木創士, 門脇多人, 萩原洋一, 多様な BYOD 機器を考慮した次世代型仮想デスクトップ (DaaS) サービスの共創, デジタルプラクティス, vol. 7, No. 2, pp. 136–147, 2016.