# 「認証統合に対応したウェブホスティングサービスの構築と運用」に至る道程

## Road to Construction and Operation of Web Hosting Service Supporting Single Sign On

豊橋技術科学大学 情報メディア基盤センター教授　土屋 雅稔
Toyohashi University of Technology, Information and Media Center, Professor, Masatoshi Tsuchiya
ORCID ID : https://orcid.org/0000-0003-1862-8149

**[紹介論文]**
認証統合に対応したウェブホスティングサービスの構築と運用
土屋 雅稔，中村 純哉，小林 真佐大，下條 詠司
学術情報処理研究 , Vol. 27, No. 1, pp. 73-81, 2023.

## 1. はじめに

現代のネットワーク社会において業務を遂行するには，安定したサーバ資源が必要不可欠である．インターネット黎明期においては，そのようなサーバ資源は，専門外の職員や学生のボランティア的活動によって維持されてきた．しかし，近年，これらのサーバの運用に興味を持ち，自発的に技術研鑽を行う職員や学生は確保困難になりつつある．また，サーバを維持するために必要な技術は高度化する一方であり，十分なメンテナンスが行われていないサーバは少なくない．

この問題に対応するため，各地の大学付属情報基盤センターでは，ホスティングサービスを提供することが広く行われている．筆者が所属する豊橋技術科学大学情報メディア基盤センターも，2008 年からウェブ・メール・DNS のホスティングサービスを提供している．ただし，豊橋技術科学大学は，学生約 2000 人の小規模単科大学であり，センターの規模も極めて小さい．そのような小規模センターにおいてホスティングサービスを実現するには，様々な工夫を必要とした．表向きの工夫については，文献 [1] [2] および紹介論文 [3] に記述済みであるから，本稿では裏向きの事情について述べたいと思う．

## 2. 第1期システム（2008年～2015年）

筆者が，情報処理センターに採用されたのは 2004 年である．翌年には，情報処理センターとマルチメディアセンターが統合され，情報メディア基盤センターに改組された．最初に課題になったのは，情報処理センターとマルチメディアセンターで別個に運用されていた教育用端末システムのアカウントの統合である．タイミング良く，情報処理センターが運用していた教育用端末システムの更新が 2006 年 3 月に予定されていた．そのユーザ認証基盤として LDAP サーバを構築するよう仕様を指定した上で，マルチメディアセンターが運用していた教育用端末システムおよび無線 LAN からは，新システムの LDAP サーバを参照するよう設定変更した．加えて，アカウント管理システムを独自に作成し，ようやく，入学時に全構成員を対象としてアカウントを配布できる体制が整った．

次に取り組んだのが，ホスティングサービスの開発である．当時は，研究室や学科で個別にウェブサーバやメールサーバを運用している体制が当たり前だったが，管理できる職員や学生の確保が困難になり，セキュリティ的な脆弱性を抱えているサーバが問題になり始めていた．アカウント管理システムが完成していなかった時期でもあり，そんな大変な仕事をやりたくなかった筆者がごねていたところ，某先生に「がたがた言わずにとっととやれ」と雷を落とされて，しぶしぶ開発を始めることになったのも，今となっては懐かしい思い出である．

第 1 期システムの開発においては，開発と運用に関わる人員は筆者ただ 1 人しかいなかったため，サービスを運用する工数を許容できる範囲に収めることが非常に重要な課題となった．他大学のサービス内容を比較検討したところ，サービスを提供するセンターが，少数の管理用アカウントを利用部局に対して発行し，利用部局の管理者は，その管理用アカウントを使って様々な作業を行うという形態が一般的だった．しかし，この形態では，管理用アカウントの引継やパスワード再設定などのサポート業務が発生することが容易に想像できたため，

管理用アカウントを使わない方法はないだろうか？と考えたことが，第 1 期システムの主題となる．

先に述べた通り，ホスティングサービスを開発する直前に LDAP サーバを用いた認証統合に着手していたため，利用部局の管理者といっても，ユーザ認証基盤上では単なる 1 人の利用者として認証が可能であることは，すぐに気がついた．この着想に基づいて，管理者の認証と管理行為の認可の分離を徹底したホスティングサービスとして実現したシステムが，第 1 期システムである．

2008 年から 2015 年まで運用した第 1 期システムの構成を，図 1 に示す．第 1 期システムでは，仮想化機構として Linux VServer[4] を採用した．Linux VServer は、FreeBSD jail とよく似たカーネルレベル仮想化機構である．各ドメインのプロセスは同一のカーネルを共有するが，ドメイン毎に異なるプロセス ID 空間で動作する．図 2 に示す通り，OS イメージやウェブサイトのデータは NFS サーバ上に用意された各ドメイン用の領域（ディレクトリ）に配置され，ドメイン毎にルートディレクトリを chroot することによって，ドメイン間のファイル名空間の分離が実現される．つまり，ドメイン分離のためにハードウェアによる仮想化支援機能を必要としないし，かつ，ドメインを切り替えるコストも極めて小さい．よって，安価な物理サーバで多数のドメインを収容することができ，厳しい予算制約を満たすために適していた．

第 1 期システムの詳細は，文献 [1] [2] に記載の通りである．

## 3. 第2期システム（2015年～2021年）

図 3 に示す通り，ホスティングサービスの利用部局は順調に拡大し，2014 年には運用スタッフが 2 人に増員された．これによって，ドメイン毎の変更履歴の記録が不十分という問題が顕在化する．第 1 期システムでは，ドメイン固有の設定変更は，当該ドメインのディレクトリツリー内のファイルを直接修正し，作業メモを WiKi に記録するという形で作業を行っていた．この形では，あるドメインの設定内容が正しいかどうかを確認するには，作業メモを全て目視で確認しなければならない．作業メモのどのあたりにどういう記述があるか，熟知していた筆者にとっては容易い確認作業も，新規に加わった運用スタッフにとっては非常に困難だったわけである．

更に，Linux VServer の運用コストが問題化する．Linux VServer を実行するには，プロセス ID 空間の分離を実現する修正パッチを適用したカーネルが必要である．第 1 期システム開発時は，上流ディストリビューション（Debian GNU/Linux）から修正パッチ適用済みカーネルが配布されていたが，第 1 期システム運用中に配布されなくなった．第 1 期システムは物理サーバ上に構築されていたため，ハードウェアの寿命によるリプレースが必要になったが，単純にハードウェアを置き換えて Linux VServer の利用を継続するという方針は運用コストの観点から困難になった．

2011 年から教育用システムの各種サーバをオンプレミス仮想化基盤に収容していたため，第 2 期システムの各種サーバも仮想化基盤上に構築することを選択した．とにかく，ホスティングサービスのための独立した予算は存在しなかったのである．ただし，ドメイン毎に別個に仮想マシンを割り当てることは，サーバ資源と運用コストの 2 つの観点から現実的ではない．そのため，仮想マシン上でコンテナ環境を実行することによって，ドメイン毎の分離を実現することにした．2015 年から 2021 年まで運用した第 2 期システムの構成を，図 4 に示す．第 2 期システムでは，コンテナ型仮想化機構として Docker[5] を採用した．Docker は Linux 標準カー

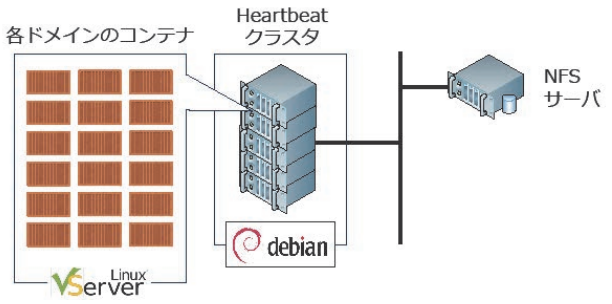

図 1　第 1 期システムの構成

図 2　第 1 期システムのデータ配置

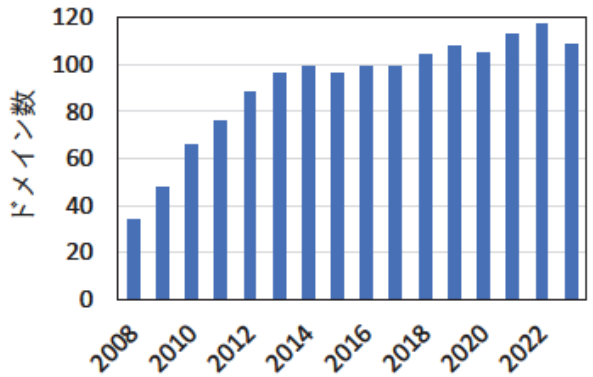

図 3　利用ドメイン数の推移

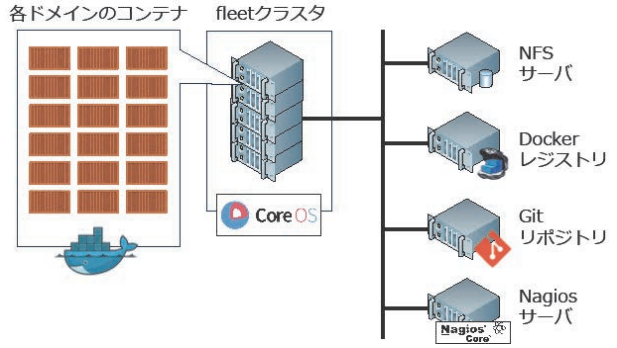


図 4　第 2 期・第 3 期システムの構成

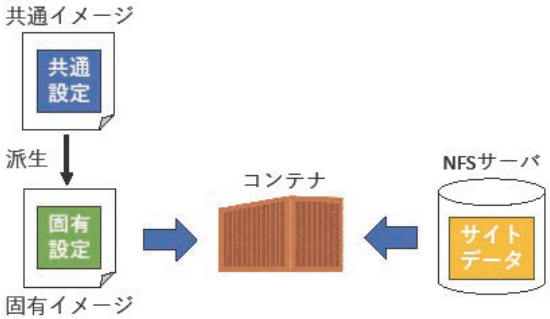


図 5　第 2 期システムのデータ配置

ネルに統合されたコンテナ実行機能に基づいて構成されており，Linux VServer で生じた運用コストの問題が発生しにくいと判断したためである．

Docker では，コンテナイメージの作成手順を Dockerfile と呼ばれるテキストファイルに記述し，それを docker コマンドに渡すとコンテナイメージを作成できる．また，既存のコンテナイメージを継承して新しいイメージを作成できるため，コンテナイメージを階層的に管理することも可能である．第 2 期システムでは，図 5 に示す通り，この仕組みを利用して，全ドメインに共通する設定と各ドメインの個別設定を分離し，かつ全体を網羅的にバージョン管理することを試みた．具体的には，全てのドメインで利用するプログラムや設定を行う Dockerfile を用意して，共通コンテナイメージを作成する．次に，各ドメインの固有設定を行う Dockerfile を用意して，ドメイン個別のコンテナイメージを作成する．その上で，全ての Dockerfile の変更履歴を Git を用いて記録すると，変更履歴が確実に保存されるという仕組みである．

## 4. 第3期システム（2021年～現在）

2018 年には運用スタッフが 3 人に増員された．これによって，各ドメインのコンテナイメージを生成する Dockerfile は，各ドメインの構成定義という観点からは不十分であるという問題が顕在化する．例として，第 2 期システムにおいて，あるドメインで MySQL サーバを利用したいという要望があった場合を考える．この場合，運用スタッフは，以下の内容を当該ドメインの Dockerfile に書き加えた上で，当該ドメインのコンテナイメージを再作成・実行する．

- MySQL パッケージをインストールする
- MySQL の設定ファイルを配置する
- MySQL サービスをコンテナ実行時に起動するよう設定する

設定後に，MySQL サーバを利用している全てのドメインに対して設定変更が必要となった場合には，全てのドメインの Dockerfile を grep して対象ドメインを列挙して対応していた．つまり，Dockerfile を参照すれば，各ドメインの構成定義も機械的に判定できるはずという考え方である．しかし，Dockerfile の記述方法は一意ではない．運用スタッフが 2 人しかいなかった時期は，記述方法のバリエーションも限定されていたが，運用スタッフが 3 人に増員されると，バリエーションが一気に増えて，構成定義が機械的に判定できない事例が続発するようになり，運用コストが急激に悪化した．

この問題を解決するため，第 3 期システムでは，各ドメインの構成定義の管理方式を図 6 のように改めた．まず，ドメイン毎の構成は，宣言的に記述した構成定義ファイルにまとめて，バージョン管理する．次に，ドメイン毎に個別のコンテナイメージを用意するのではなく，全てのドメインが共通のコンテナイメージを用いる．この共通コンテナイメージは，コンテナ実行時に構成定義ファイルを読み込み，必要な設定をコンテナに施す．この方式によって各ドメインの構成が明確になり，また設定変更も容易になった．

近年，セキュリティ上の要請に基づき，一般に広く使われているウェブブラウザ（Google Chrome や Apple Safari など）は，TLS 化されていないウェブサーバにアクセスすると警告を出力したり，アクセスを拒否するようになっている．そのため，第 2 期システムにおいては，TLS 化が必要な一部のドメインだけを対象として，UPKI によって発行された証明書を設定して対応していたが，第 3 期システムにおいては，全てのドメインを対象として TLS 化する必要がある．さらに，サーバ証明書の有効期限短縮化によって，証明書の更新作業がきわめて頻繁に発生するため，更新の自動化も重要である．以上より，第 3 期システムにおいては，

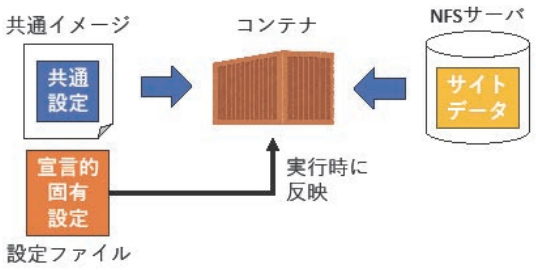


図６　第３期システムのデータ配置

Let's Encrypt によって発行された証明書を基本的に用いることにした．UPKI によって発行される証明書は OV 証明書であるのに対して，Let's Encrypt によって発行される証明書は DV 証明書であるから，証明書の信頼性という観点では UPKI の方が好ましい．しかし，UPKI の証明書発行システムは自動化に対応していないため，第３期システムでは採用できなかった．

第２期システムおよび第３期システムの詳細は，紹介論文 [3] に記載の通りである．

## 5. おわりに

本稿は，「小規模センターにおいて，10 年以上の長期にわたってホスティングサービスを提供できた秘訣は何だろうか」という質問を編集委員氏から受けて書き始めたものである．本サービスの運用経験を振り返ってみると，運用スタッフの変化がトリガーとなって問題が顕在化し，その問題を解決するためにシステムのリファクタリングを行い，そのプロセスを論文化するという過程を繰り返していることが分かる．おそらく，リファクタリングをきちんと実践できたことが，サービスを維持できた秘訣ではないだろうか．とすると，そのような実運用の知見を投稿できる場所として「学術情報処理研究」が用意されていたことは，大きな助けになっていたと考えられる．

【著者略歴】

土屋 雅稔

2004 豊橋技術科学大学情報処理センター助手．2007 同大学情報メディア基盤センター助教．2014 同大学情報メディア基盤センター准教授．2023 同大学情報・知能工学系教授（情報メディア基盤センター兼務）．自然言語処理に関する研究に従事．博士（情報学）．言語処理学会，情報処理学会，人工知能学会 各会員．

関係者の尽力に感謝するとともに，実運用に関する知見を広く扱う論文誌として存続されることを期待する．

さて，運用スタッフの変化がシステムのリファクタリングを要請するという経験則が正しいならば，近い将来，第４期システムが必要になることが予想される．と言うのも，筆者の異動に伴って，本サービスに対する筆者のエフォートが低下しつつあるからである．どのような第４期システムが実現されることになるか，今から楽しみである．

最後に，ホスティングサービスの開発と運用に協力してくださったセンタースタッフの皆様に，深く感謝します．また，サービス開発のきっかけを作ってくださった廣津登志夫先生（元豊橋技術科学大学情報メディア基盤センターネットワーク部長．法政大学教授），知見を論文化するよう励まし続けてくださった稲垣康善先生（元豊橋技術科学大学副学長兼情報メディア基盤センター長．名古屋大学名誉教授）に感謝します．

## 参考文献

[1] 土屋雅稔：「認証基盤と連携したメールホスティング環境の構築」，学術情報処理研究，Vol.13, pp.5-16（2009）

[2] 土屋雅稔：「管理者が安全に交代できる学内ホスティングサービス」，電子情報通信学会論文誌，Vol.J95-B，No.10, pp.1264-1272（2012）

[3] 土屋雅稔，中村純哉，小林真佐大，下條詠司：「認証統合に対応したウェブホスティングサービスの構築と運用」，学術情報処理研究，Vol.27, pp.73-81（2023）

[4] B. Des Ligneris: Virtualization of Linux based computers: The Linux-VServer project. Proceedings of the 19th International Symposium on High Performance Computing Systems and Applications（HPCS'05）, pp.340-346（2005）

[5] Dirk Merkel: Docker: lightweight linux containers for consistent de velopment and deployment. Linux Journal, Vol.239, No.2, pp.2（2014）