Open OnDemand の導入と Keycloak による多要素認証の実現

戸田 庸介 [1)], 當山 達也 [1)], 島袋 友里 [1)], 疋田 淳一 [1)]

1) 京都大学 情報部

toda.yosuke.4e@kyoto-u.ac.jp

Deployment of Open OnDemand and multi-factor authentication with Keycloak

Yosuke Toda [1)], Tatsuya Tohyama [1)], Yuri Shimabukuro [1)], Junichi Hikita [1)]

1) Information Infrastructure Division, Kyoto Univ.

概要

京都大学学術情報メディアセンターでは，これまでスーパーコンピュータへのアクセスにおける利便性とセキュリティの両立を課題としてきた．従来の公開鍵認証やコマンドライン操作がもたらす利用者負担を解消するため，Web ベースの GUI ポータルである Open OnDemand を導入し，さらにオープンソースのシングルサインオン認証サーバーである Keycloak と連携させることで，多要素認証を必須化した新しいアクセス環境を構築した．本稿では，システムの設計，特に Open OnDemand によるアプリケーションの機能拡張および，Keycloak を中心とした認証基盤の詳細な構築プロセスと運用の取り組みについて報告する．

1 はじめに

京都大学学術情報メディアセンター(以下，本センターという) では，全国共同利用設備として，スーパーコンピュータシステム（以下，システムという）を運用し，学内外の研究者へ計算機資源を提供している．システムへの接続方法としては公開鍵認証を用いた SSH 接続と，Web ブラウザ上でコンソール操作が可能な FastX という Web アプリケーションを用いた接続の 2 つの方法を提供している．

FastX を用いてシステムを利用する場合でも，公開鍵認証を必須としているため，利用者は鍵ペアを作成し，システムに登録する必要がある．しかし，卒業研究で初めて利用する学生や，システムで提供しているアプリケーションを利用することが目的の利用者にとっては，鍵ペアの作成は最初の障壁となりやすい．また，コマンドラインによるアプリケーションの起動や，ジョブ投入などの操作に関する問い合わせが多くあり，初級者が簡単に使えるシステムの検討が必要となった．

そこで本稿では，「富岳」を運用する理化学研究所をはじめとして，国内外の大学でも導入が進んでいる Open OnDemand について，本センターにおける導入に向けた検討から実装までの取り組みとその評価について述べる．

2 本センターのシステムと課題

2.1 既存システム構成

本センターのシステムは，図 1 に示す通り Camphor3，Laurel3，Cinnamon3，Gardenia，クラウドシステムからなる 5 種の演算システムと，大容量ストレージ，高速ストレージからなる 2 種のストレージシステムによって構成された，総演算性能 11PFLOPS，総メモリ容量 370TiB，総ストレージ容量 44 PB の高性能かつ大規模なシステムである[1].

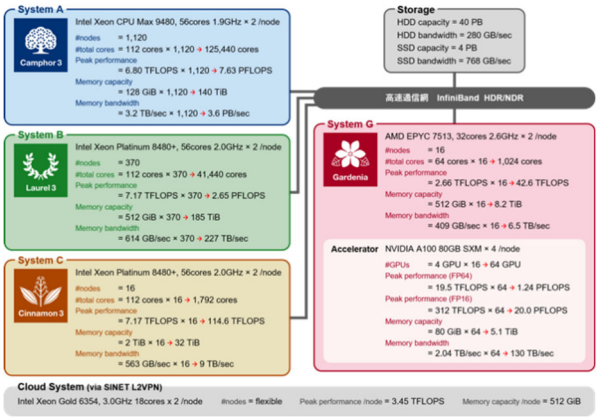

図 1 システム構成図

2.2 従来の接続方法と課題

システムへの接続には，これまで主に SSH 接続と FastX 経由の Web アクセスを提供してきた．

SSH 接続は，標準的なコマンドラインアクセスであり，公開鍵認証に限定する形で利用者への提供を行っている．これに加えて，2020 年 6 月には，コマンドライン操作や GUI アプリケーションを Web ブラウザで実行することが可能な FastX という Web アプリケーションを導入し，主に GUI アプリケーション利用時の操作を簡素化している．なお，FastX についても SSH 接続と同様に公開鍵認証に限定して，利用者への提供を行っている．

FastX の導入により，GUI アプリケーションを利用するための設定に関する問い合わせは減少したが，アプリケーションの起動方法についての問い合わせは依然として多い．また，鍵ペアの作成方法や，ジョブ実行時の資源確保の方法など，利用開始時の基本操作に関する問い合わせについても依然として多い状況であり，利用者が円滑にシステムの利用を開始する上での課題となっている．

3 Open OnDemand の導入

3.1 導入背景

2.2 節で述べたように，FastX の導入により GUI 環境の利用は改善されたものの，問い合わせの状況から，鍵ペアの作成やジョブ実行といった利用開始時の基本操作に関する課題があると考えられた．これらの課題を根本的に解決するためには，より直感的で技術的な前提知識をあまり必要としないアクセス手段の提供が必要である．

そこで本センターでは， Web ブラウザを介してシステムにアクセスすることが可能な Open OnDemand に注目した．Open OnDemand は，利用者が Web ブラウザを介してシステムにアクセスし，Web ブラウザ上でバッチジョブの投入やアプリケーションの起動，ファイル管理などの作業を直感的に実行することが可能なオープンソースの Web アプリケーションである[2][3]．これを用いることで，2.2 節で述べた課題の解決が期待できると考えた．

さらに， 近年の AI 研究の進展に伴い，Jupyter Notebook のような Web ベースの対話型開発環境の需要が急速に高まっている．従来の環境では，計算ノード上で Jupyter Notebook を起動し，適切なポート転送設定を行った上でローカルの Web ブラウザから接続する必要があり，これらの一連の作業は初学者にとって技術的なハードルとなっていた．Open OnDemand では，このような複雑な設定作業を自動化することが可能である．利用者が簡単な操作で Jupyter Notebook を利用できる環境を提供することで，利用に必要な学習コストを大幅に削減できると期待されることも，導入に際しての重要な背景の一つである．

3.2 導入方針

Open OnDemand の導入にあたり，本センターでは，2.2 節で述べた課題を効果的に解決するために，以下の設計思想に基づいて検討を進めた．

まず，複数クラスタの統合的利用環境の構築を目指した．従来の SSH 接続と FastX では，各クラスタに個別のログインノードが設置されており，利用者は利用するクラスタごとに接続設定を行う必要があった．Open OnDemand では，本センターが運用する 4 つのクラスタ（Camphor3，Laurel3，Cinnamon3，Gardenia）を単一の Web ポータルから統一的にアクセスできる環境を構築することで，クラスタ間での利用方法の違いを意識することなく，シームレスな利用環境の提供を目指した．

次に，対話型開発環境の利用支援について検討した．利用者からも要望が多く挙がっていた Jupyter Notebook については，従来必要であったポート転送設定やコマンドライン操作を排除し，Web ブラウザから直接起動し，利用できる環境を整備することで，データ解析や機械学習分野における研究活動の効率化を図った．

また，GUI アプリケーションの操作性向上も重要な目標とした．FastX により一定の改善は図られているが，さらなる使いやすさの向上を目指し，

より直感的な操作で GUI アプリケーションを利用できる環境を提供することにより，技術的な操作に習熟していない利用者でも容易に利用できる環境の実現を目指した．

3.3 Open OnDemand の実装

3.3.1 基本実装

Open OnDemand の導入にあたり，複数台あるログインノードのうち 1 台を Open OnDemand 専用機として切り離し，Open OnDemand 本体および関連モジュールを導入した．また，計算ノード上で起動した GUI アプリケーションを操作するために，各計算ノードに TurboVNC と websockify を導入した．これにより，計算ノードの TurboVNC で起動した仮想デスクトップを websockify で WebSocket プロトコルに変換し，Web ブラウザ上で操作することが可能となった．

2.1 節で説明した通り，本センターの学内にあるシステムは，4 つのクラスタによって構成されているが，クラスタごとに Open OnDemand を起動することは効率的ではなく，また利用者にとっても混乱の元となることから，1 台の Open OnDemand サーバーから，すべてのクラスタへのジョブを一元的に投入することができるような設定とした．

3.3.2 アプリケーションの整備

Open OnDemand には，標準的な機能として，ファイル操作，ジョブ投入，ジョブの実行状況確認，シェルアクセスなどが搭載されている．

本検討では，直感的で技術的な前提知識をあまり必要とすることなく，システムを利用可能な環境の構築が重要であり，導入においては，特に要望が多くあった Jupyter Notebook や，GUI アプリケーションの利用を支援する必要がある．そこで，Jupyter Notebook に加え，本センターのシステムで利用することが可能な 15 の GUI アプリケーションを簡易に利用できるように設定した．

また，Open OnDemand 上で設定したアプリケーションについては，図 2 のように，ダッシュボードと呼ばれる Open OnDemand のトップページ上でカテゴリごとに表示することで，利用者が目的のアプリケーションを直感的に探し出せるようになり，利便性の向上を実現した．

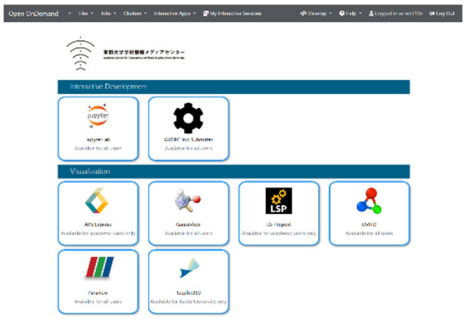

図 2 ダッシュボードに並ぶ GUI 操作系アプリケーション群

3.3.3 ジョブ投入支援機能の開発

Open OnDemand にはジョブ投入を支援するアプリケーションとして Job Composer が標準で搭載されているが，初学者にとっては操作が複雑で理解しにくいという課題がある．そこで，初学者でも簡易にジョブ投入を支援するアプリケーションとして，KUDPC Job Submitter を開発した．

KUDPC Job Submitter は，手元で作成したジョブスクリプトをアップロードしてジョブを投入する機能や，Web フォーム上でジョブスクリプトを作成し実行する機能を備えている．Web フォームは，Open OnDemand のフレームワークを活用して作成し，JavaScript による動的制御を組み込んだ．その結果，図 3 と図 4 に示すような直感的操作が可能なインターフェースを実現した．

これらの実装により，従来のコマンドライン操作や複雑な設定を必要とせず，Web ブラウザから直感的にシステムを利用できる環境を実現した．特に，初学者が最初につまずきやすい公開鍵認証の設定やジョブスクリプトの記述といった部分における障壁を大幅に低減することができると期待される．

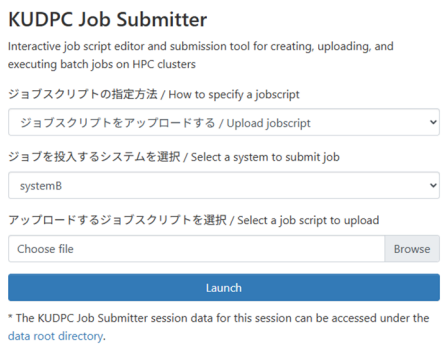

図 3　ジョブ投入支援機能のアップロード画面

図 4　ジョブ投入支援機能の編集画面

4 セキュリティ強化への取り組み

4.1 セキュリティの課題

Open OnDemand の導入により，利用者の利便性は大幅に向上する見込みとなったが，一方で従来のシステムへのアクセス方式とは異なることから，セキュリティ上の課題について検討する必要がある．従来の SSH や FastX によるシステムへの接続では，公開鍵認証による認証機構を提供していたが，Open OnDemand では ID・パスワードによる認証が標準となっており，これだけでは不正アクセスのリスクが増大すると考えられる．なお，Open OnDemand 自体には公開鍵認証による認証機構が標準で搭載されていないが，認証認可の仕組みを外部のシステムと連携させることができる．利便性の向上により新規利用者の増加が予想される中，セキュリティ意識の異なる多様な利用者層に対しても適用可能で，かつ管理負荷を増大させない認証方式として，従来の公開鍵認証に代わる多要素認証システムの導入が必要であると考えた．

4.2 多要素認証システムの検討

Open OnDemand は OpenID Connect プロトコルによる外部認証システムとの連携機能が用意されており，この機能を活用することにした．OpenID Connect 対応の認証システム構築にあたり，社会的に広く普及している TOTP（Time-based One-Time Password）による多要素認証機能が標準で提供されていることに加え，自由に認証方式を追加することが可能なフレームワークが提供されており，かつ効率的に機能を実装できることを選定条件とした．

これらの要件を満たすソリューションとして，本センターでは，オープンソースの認証・認可サーバーである Keycloak を採用した．Keycloak は OpenID Connect および SAML プロトコルに対応しており，プラグインアーキテクチャによる機能拡張が容易である．

図 5 に，認証システムも含めたシステム全体の構成イメージを示す．

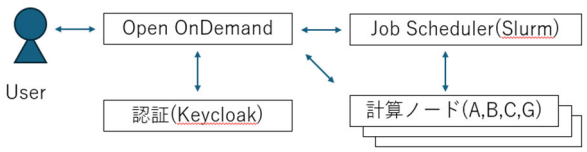

図 5　システム全体の構成イメージ

4.3 多要素認証システムの設計

TOTP 認証の導入にあたり，利用者の利便性とセキュリティのバランスを考慮した認証フローの設計を行った．Keycloak の標準機能では，初回ログイン時に ID・パスワード認証後，自動的に TOTP デバイスの登録プロセスに進む仕組みとなっている．

しかし，この方式では，初回アクセス時に第三者による TOTP デバイス登録のリスクが存在する．本センターでは，アカウント申請時にメールアドレスによる本人確認を実施しているため，この確認済みメールアドレスを活用した追加の本人確認手順を導入することとした．

具体的には，初回ログイン時およびデバイス紛失時の代替手段として，メール OTP（One-Time Password）による本人確認を実施し，その後 TOTP デバイスの登録を行う段階的な認証フローを設計した．通常のログイン時は TOTP 認証のみで完結し，特別な状況でのみメールによる OTP を使用することで，利便性を保ちながら従来の公開鍵認証と同等のセキュリティレベルを担保する方針とした．

4.4 メール OTP 認証プラグインの開発

4.3 節で設計したメール OTP 認証機能を実現するため，Keycloak の SPI（Service Provider Interface）を活用した独自プラグインの開発を行った．SPI は Keycloak が提供する拡張機能であり，認証フローやイベントリスナーなどの機能を， Keycloak 本体のソースコードに手を加えることなく，プラグインとして追加できる仕組みである．

メール OTP 認証プラグインは，Keycloak の Authenticator SPI を利用する形で開発した．このプラグインは，認証コードの生成・送信を行う authenticate メソッドと，利用者が入力したコードの照合を行う action メソッドの 2 つの主要な処理で構成される．

authenticate メソッドでは，まず利用者のセッション情報からメールアドレスを取得し，乱数生成機能を用いて 6 桁の数字からなる認証コードを生成する．生成された認証コードは，有効期限とともにサーバー側のセッションストレージに保存され，同時に利用者のメールアドレス宛にテンプレート化されたメールとして送信される．メール送信には Keycloak が提供するメール送信機能を活用し，本センターのメールサーバーとの連携を行う．

action メソッドでは，利用者が Web フォームに入力した認証コードを受け取り，セッションストレージに保存されている正しいコードとの照合を行う．照合時には，コードの一致性に加えて有効期限内であることを確認し，いずれかの条件を満たさない場合は認証を拒否する．認証成功時には，セッションストレージから該当する認証コード情報を削除し，次の認証ステップに進む流れとなる．

4.5 認証フローの設計と実装

開発したメール OTP 認証機能を，Keycloak が標準で提供する TOTP 認証機能と統合するため，認証フローの詳細な設計を行った．Keycloak では，Authentication Flow と呼ばれる仕組みにより，複数の認証ステップを組み合わせた複雑な認証シーケンスを定義することができる．

本システムでは，利用者の状況に応じて 3 つの認証パターンを実装した．第一に，通常ログイン時の認証フローでは，ID・パスワード認証に続いて TOTP 認証を実行し，両方が成功した場合にのみシステムへのアクセスを許可する．第二に，初回ログイン時の認証フローでは，ID・パスワード認証後にメール OTP 認証を実行し，本人確認が完了した後に TOTP デバイスの登録画面に遷移する．第三に，TOTP デバイス紛失時の代替認証フローでは，ID・パスワード認証後にメール OTP 認証を実行し，認証成功後に TOTP デバイスの再登録を可能とする．

これらの認証フローは，利用者の TOTP 登録状態やアクセス時の状況を自動的に判定し，適切なフローを選択する仕組みとした．具体的には，利用者のプロファイル情報から TOTP デバイスの

登録有無を確認し，未登録の場合は初回登録フローに，登録済みかつ通常アクセスの場合は標準フローに, TOTP 認証に失敗した場合や, 利用者がデバイス紛失を申告した場合は，代替フローにそれぞれ自動的に分岐する．この実装により，従来の公開鍵認証と同等のセキュリティレベルを持ちながら，利用者の利便性を大幅に向上させる多要素認証システムを実現した．本件で実装した多要素認証フローを図 6 に示す.

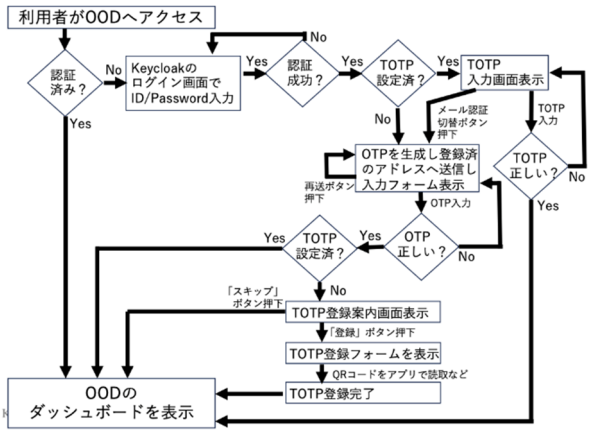

図６ 多要素認証フロー図

5 評価

2025 年 6 月 13 日に試行サービスとして Open OnDemand の提供を開始し，本センター主催のアプリケーション利用に関する講習会の一部において，システムへの接続方法を従来の FastX から Open OnDemand に変更して開催した.

受講者には専用アカウントを発行しているが，管理者側の視点では，従来必要であった公開鍵の事前準備作業が完全に不要となった. これにより，講習会準備における管理負荷が大幅に軽減された. また，受講者側の視点では，従来必要であった本人確認情報と講習会開始時に提示される認証コードを用いた鍵ペアのダウンロード，および FastX の初期設定手続きが不要となった. 代わりに，ID・パスワードおよび講習会参加申込時に登録したメールアドレスに送信されるメール OTP による認証のみで，ログインすることが可能となった.

一方で，いくつかの運用上の課題も明らかになった．メール OTP が迷惑メールフォルダに振り分けられる事例や，参加者のネットワーク環境の制約により Open OnDemand 自体にアクセスできない事例が発生した．これらの問題については,事前の案内方法の見直しや参加者への詳細な接続手順の提供などにより改善を図る必要がある.

6 まとめと今後の課題

本稿では，本センターにおいて Open OnDemand を導入し，Keycloak による多要素認証システムを用いることで，システムへのアクセスにおける利便性とセキュリティの両立を実現した取り組みについて報告した．2025 年 9 月時点でのユニークな利用者数は 31 名であり, 本センターの利用者数の 5%未満の利用であることから，今後の広報活動などを通して，利用拡大を図る予定である.

今後の課題としては，利用者からの要望に応じて，利用可能なアプリケーションを継続的に追加していく必要がある．また，さらなる利用促進と利便性の向上に向けて，情報提供機能の改善が重要である．現在，障害情報やメンテナンス情報といった重要な情報はヘルプメニューからのリンクで提供しているが，これらをダッシュボードにおいてウィジェットにより表示することで，利用者が必要な情報を一目で把握できる環境の構築を検討している.

謝辞 Open OnDemand の導入にあたりご尽力いただいた NEC エンジニアの方々に感謝します.

参考文献

[1] 深沢圭一郎, 新スーパーコンピュータシステムのご紹介，京都大学情報環境機構広報誌「Info!」, Vol. 28, p10-12, 2023.
[2] 中尾昌広,三浦信一,山本啓二.スーパーコンピュータ「富岳」における HPC クラスタ用 Web ポータル Open OnDemand の導入.情報処理学会研究報告, 2022-HPC-186, No. 5, 2022.
[3] Open OnDemand, https://openondemand.org