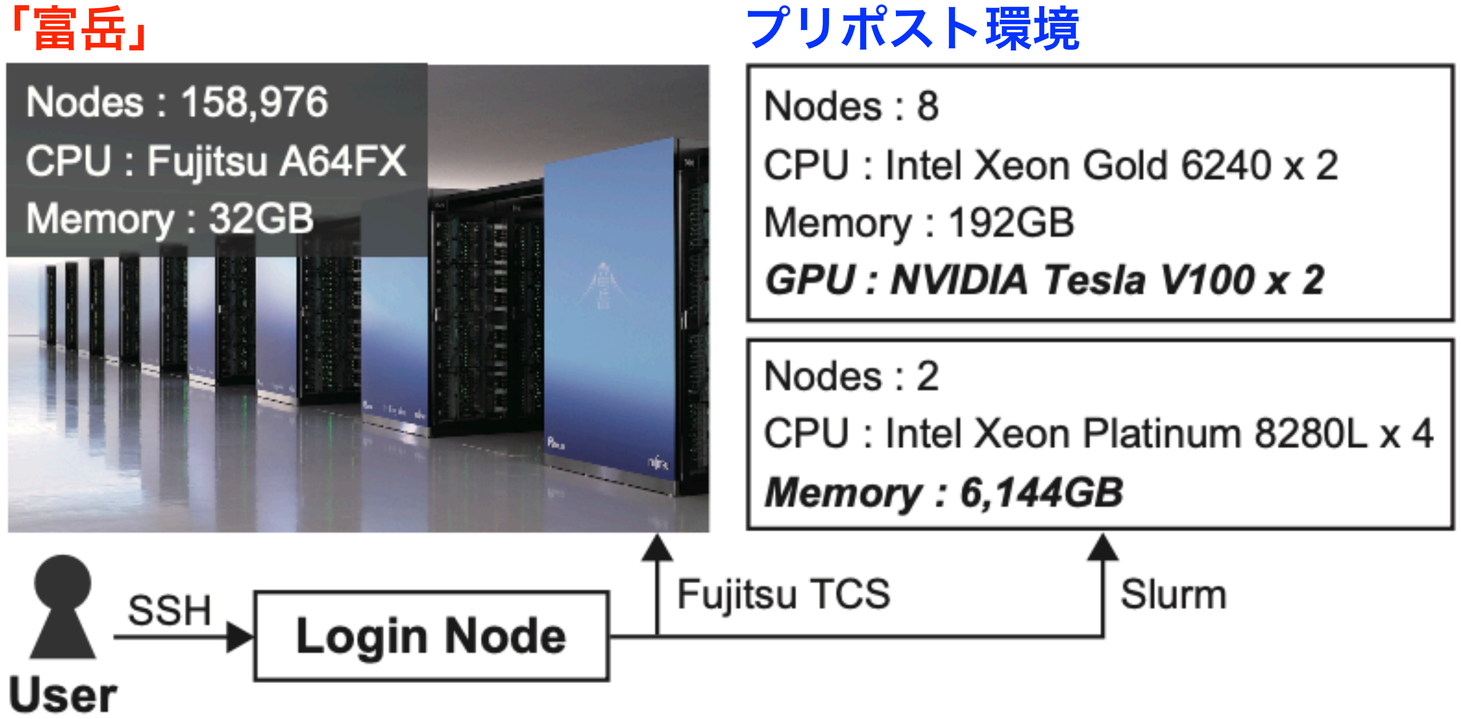
スーパーコンピュータ「富岳」における HPCクラスタ用WebポータルOpen OnDemandの導入

中尾昌広、三浦 信一、山本 啓二（理化学研究所 計算科学研究センター）

背景（1/2）

- 理化学研究所 計算科学研究センターは**「富岳」**を開発・運用
- 利便性を向上させるため、可視化やデータ変換等を行う**プリポスト環境**も提供
  - GPUを搭載した計算ノード
  - 大容量メモリを搭載した計算ノード
- 利用方法：共通のログインノードにSSHで入り、ジョブスケジューラを通してジョブを投入
- ジョブスケジューラ
  - 「富岳」は**Fujitsu TCS** (Fujitsu Software Technical Computing Suite)
  - プリポスト環境は**Slurm**



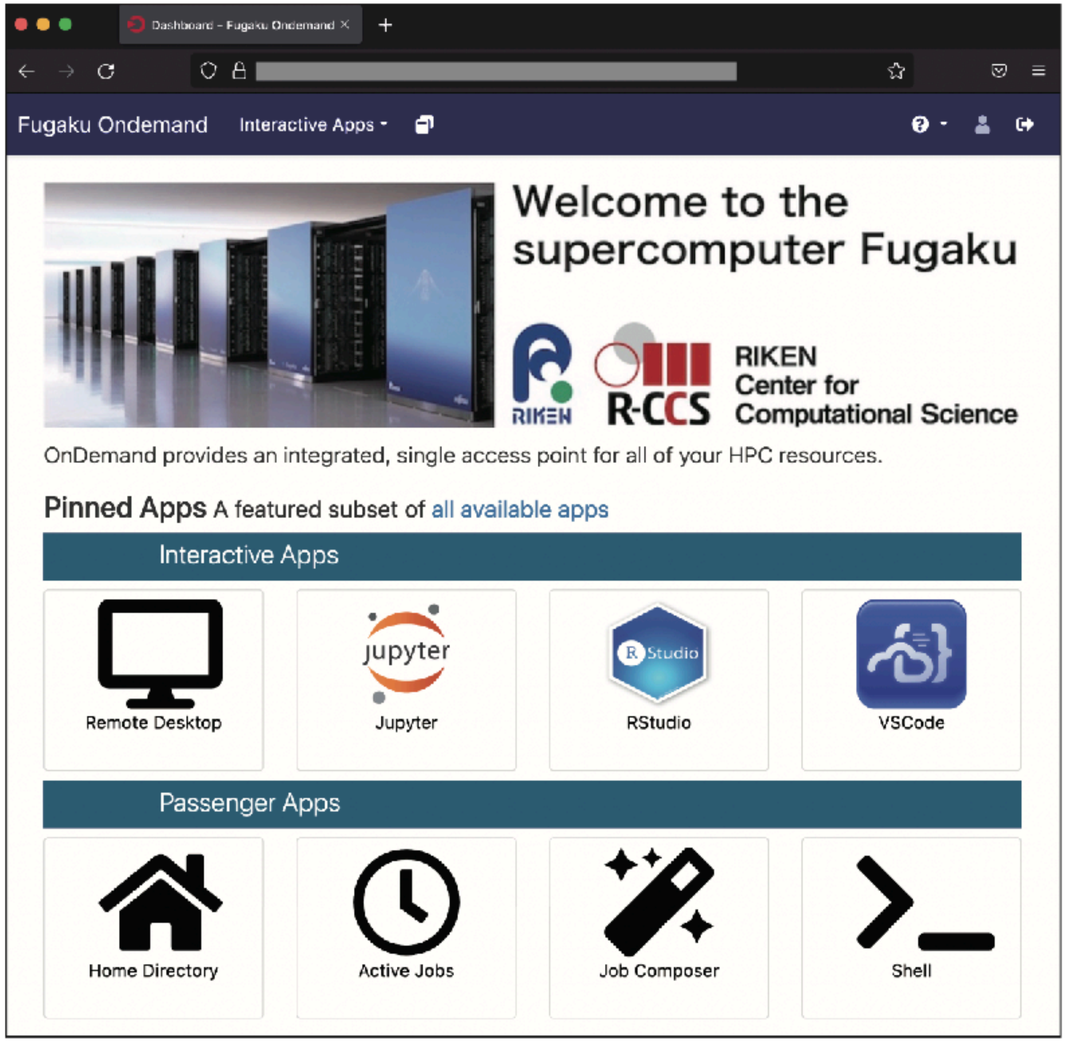
背景（2/2）

- 「富岳」などのHPCクラスタの問題点として、HPCクラスタを用いるための前提知識が多いため、初心者にとって利用するまでの学習コストが大きい
  - シェルなどのコマンドラインインタフェース
  - SSH鍵ペアの生成と公開鍵の登録
  - ジョブスケジューラ
- 対話的操作を伴うGUIアプリケーションを計算ノード上で動作させたいが、その手順は煩雑
  - 例：JupyterLabの場合
    1. SSHでログインノードにログイン
    2. ジョブスケジューラを通してJupyterLabを計算ノード上で実行
    3. 計算ノードのIPアドレスを取得
    4. 取得したIPアドレスとJupyterLabのポート番号にSSHトンネリングでローカルポートと接続
    5. ローカルポートをWebブラウザに入力

目的

- HPCクラスタ用のWebポータルOpen OnDemandを「富岳」とプリポスト環境に導入
  - SSHではなくWebブラウザからHPCクラスタを利用
  - GUIアプリケーションを計算ノード上で簡易に利用
- Open OnDemandは様々なスケジューラ（Slurmなど）に対応しているが，Fujitsu TCSには未対応

Fujitsu TCS用のアダプタの開発



Open OnDemandとは https://openondemand.org

- Open OnDemandはHPCクラスタを簡易に利用するためのオープンソースソフトウェア
- Ohio SupercomputingのWebポータルをベースに開発
- HPCクラスタの計算資源への簡易な利用手段を提供することが目的
  - 従来のSSHを用いた方法では最初のログインからジョブを投入するまでの時間の中央値が約22時間であるのに対し，Open OnDemandを用いた方法では約2時間になった[1]

[1] Settlage, Robert et al. Open ondemand: Hpc for everyone. ISC High Performance International Workshops, Frankfurt, Germany, June 16-20, 2019, Revised Selected Papers, pp. 504–513, 2019

- 多くのHPCクラスタのセンタで利用されている（日本では未確認）

Open OnDemandの機能

- Webブラウザを用いたGUIによるHPCクラスタの利用
- Interactive Apps（計算ノードで起動）
  - リモートデスクトップ環境
  - Webベースのアプリケーション（JupyterLabなど）
- Passenger Apps（Open OnDemandサーバで起動）
  - ファイル管理
  - ジョブ管理
  - ジョブ投入
  - コマンドラインによるシェルアクセス
- これらのアプリケーションは追加・削除が可能
- Open OnDemandが提供するフレームワークを用いることで，新しいアプリケーションの開発も可能

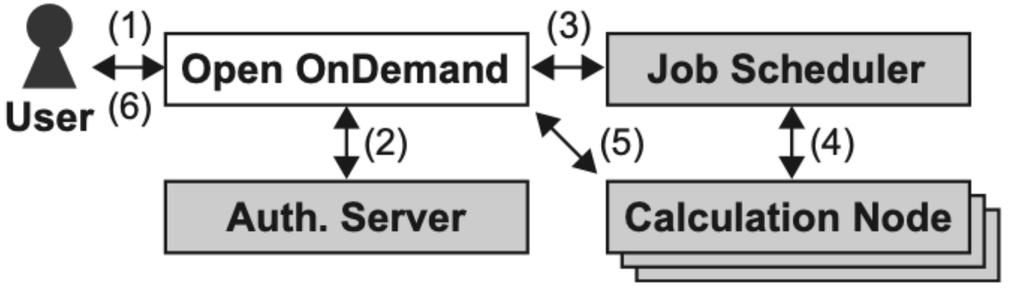
動作が確認されているInteractive Apps

- Remote Desktop
- JupyterLab
- RStudio
- VSCode
- Abaqus
- ANSYS
- COMSOL
- MATLAB
- QGIS
- Paraview
- STATA
- Tensorboard

VNCもしくはWebベースのアプリケーションは、基本的に動作するはず

Interactive Appsの動作フロー

1. Webブラウザを用いてOpen OnDemandサーバにログイン
2. ログインのためのユーザ認証
3. アプリケーションの実行命令が発行されると，計算ノードにジョブを投入
4. ジョブが実行されるまで待機
5. ジョブの実行時に，計算ノードのIPアドレスなどの情報をOpen OnDemandサーバに送信し、リバースプロキシを設定
6. リバースプロキシ用のURLを用いて，WebブラウザからHPCシステム内部の計算ノードに接続

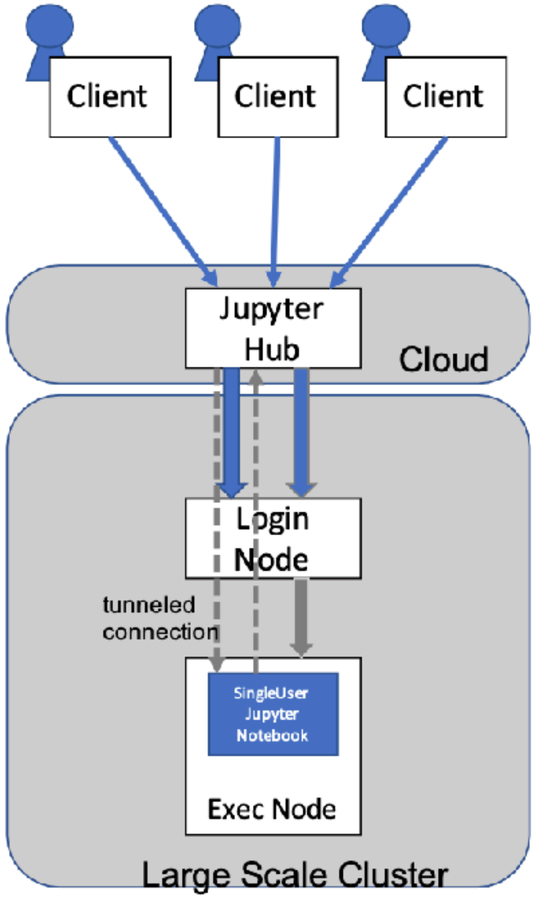


認証サーバ・ジョブスケジューラ・計算ノードはHPCクラスタが提供するサービス

既存研究との比較（1/2）

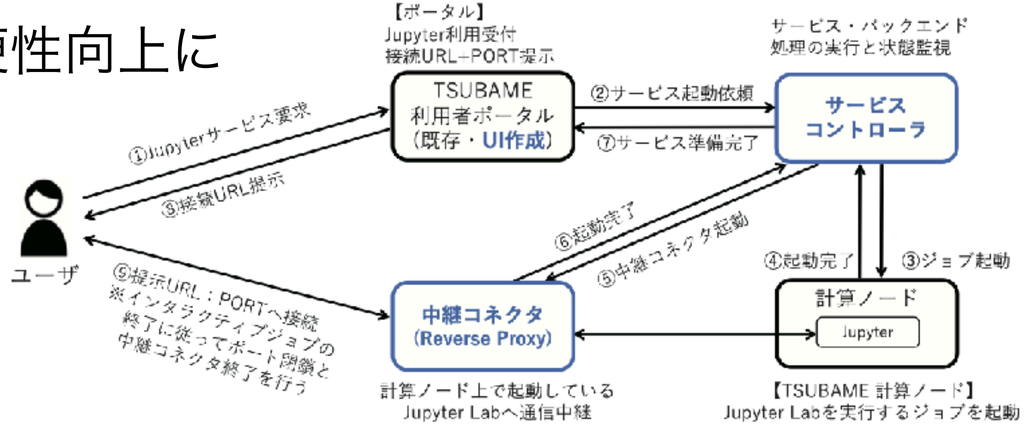
- 中田 秀基 他，大規模計算クラスタにおけるユーザ利便性向上，第174回HPC研究会，2020年5月
- Jupyter NotebookをHPCクラスタ（ABCI）の計算ノード上で動作させる方法を2つ提案

1. 前述したSSHトンネリングを自動化するスクリプト
   - ユーザの計算機環境とサーバにスクリプトをインストールし，
     そのスクリプトを手動で実行する必要がある
2. HPCクラスタとは別のクラウドサービス（AWSなどのパブリック
   クラウド）上に用意したJupyterHubを用いる
   - HPCクラスタの1つのアカウントを複数のユーザで共有することを
     前提としているため，運用ポリシによっては許可できない場合がある



既存研究との比較（2/2）

- 野村 哲弘 他，Tsubame3のインタラクティブ利用の利便性向上にむけた取り組み，第175回HPC研究会，2020年7月
- HPCクラスタの計算ノード上でJupyterLabを動作させるためのWebアプリケーション実行基盤の開発
- 動作フローはOpen OnDemandとほぼ同じ



- このWebアプリケーション実行基盤はオープンソースソフトウェアではなく，また利用可能な認証サーバ・ジョブスケジューラの種類・JupyterLab以外のアプリケーションへの適用については制限がある

Open OnDemandはオープンソースソフトウェアであり、どのような環境にも柔軟に対応するため、様々な認証サーバ・ジョブスケジューラに対応しており、アプリケーションの追加も容易に行うことが可能である

Fujitsu TCSへの対応（1/2）

- Open OnDemand-2.0.27（2022年6月リリース）が対応しているジョブスケジューラ
  - Slurm，PBS Pro，Torque，Sun Grid Engine，Univa Grid Engineなど
- Fujitsu TCSに対応するためのコード変更 & Pull Request
  1. https://github.com/OSC/ood_core/pull/766
  2. https://github.com/OSC/ondemand/pull/2194
- Open OnDemand-2.0.28（2022年8月リリース）からFujitsu TCSもサポート
  - リリースのタイミングの関係で、1. だけマージしてもらった
  - 1. だけでも、Fujitsu TCSは動作する
  - 2. はジョブの詳細表示に関するコードで、2.0.29で利用可能の予定
- Fujitsu TCSをジョブスケジューラに採用しているHPCクラスタ（東京大学のWisteria/BDEC-01，名古屋大学の不老，九州大学のITOなど）では，そのままOpen OnDemandを利用可能

Fujitsu TCSへの対応（2/2）

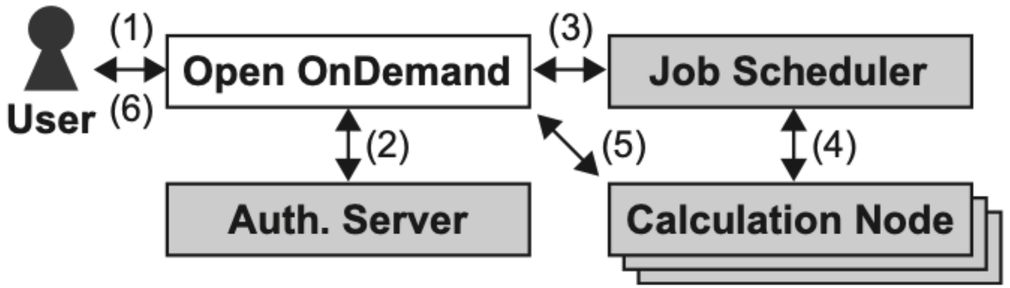
- 様々なジョブスケジューラに対応するためのアダプタインタフェースが定義
- そのインタフェースで定義されている下記のメソッドをFujitsu TCS用に実装
- 実装言語はRubyで、コード量は約400行（詳細は予稿集かPull Requestを参照ください）

submit	ジョブの投入
delete	ジョブの削除
status	ジョブの状態を取得
hold	ジョブのホールド
release	ホールドされたジョブの開放
info	ジョブの情報を取得
info_all	全ジョブの情報を取得
cluster_info	HPCクラスタのシステム情報を取得
supports_job_arrays	バルクジョブのサポートの可否
directive_prefix	ジョブスケジューラで用いられる接頭辞を取得（#PJM）

従来のログイン手順との違い

- 「富岳」の従来のログイン手順
  1. クライアント証明書をWebブラウザにインストール
  2. Webブラウザで「富岳」の利用者ポータルにログイン
  3. （SSH鍵ペアがなければ） ローカルPCでSSH鍵ペアを作成
  4. SSHの公開鍵を「富岳」の利用者ポータルに登録
  5. ログインノードにSSHでログイン

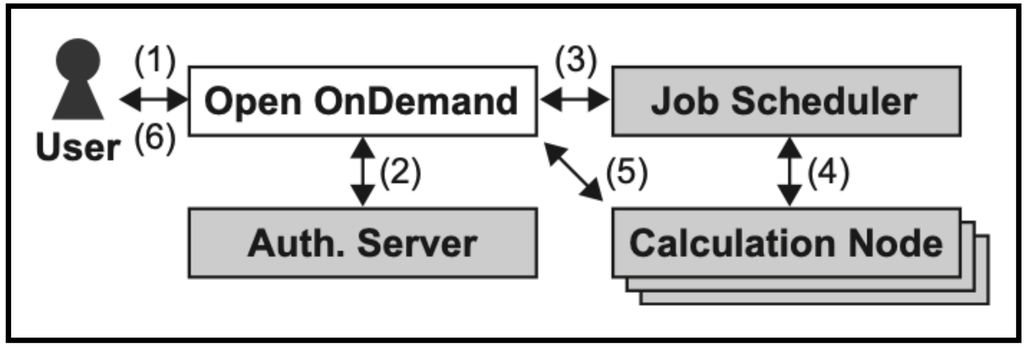
- 「富岳」のOpen OnDemandを用いたログイン手順
  1. クライアント証明書をユーザのWebブラウザにインストール（上と同じ）
  2. WebブラウザでOpen OnDemandサーバにログイン



Web上におけるシングルサインオンを実現するためのソフトウェアKeycloakを認証サーバに利用

Interactive Appsの準備

- Singularityを利用して、計算ノード上で動作させるアプリケーションを用意する
- Remote Desktop、JupyterLab、RStudio、VSCode
- 「富岳」とプリポスト環境はアーキテクチャが異なるため、それぞれのコンテナイメージを構築（計8つ）
  - 各定義ファイルはほぼ同じであるが、一部のアプリケーションのARMアーキテクチャ版のRPMパッケージが存在しなかったので、定義ファイル中で該当アプリケーションのソースコードを取得・コンパイルしている（例：Remote Desktopで用いるTurboVNCなど）
  - 定義ファイルなど：https://github.com/RIKEN-RCCS/ondemand_fugaku

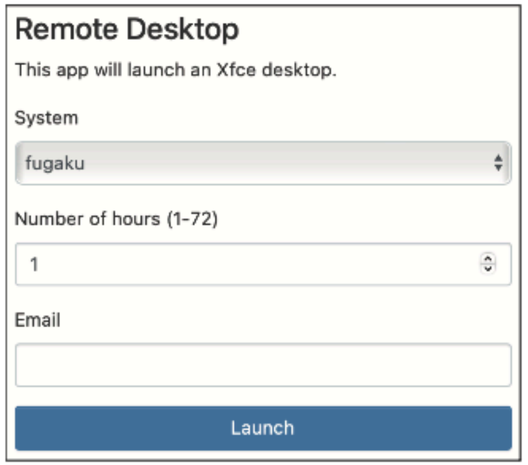


Webフォームの動的変更

- 「富岳」とプリポスト環境のジョブスケジューラの各リソースグループと設定項目の最大値は下表の通り
- プリポスト環境ではoversubscribeが可能であるため、CPUコア数・メモリ量・GPU数を指定する
- 「富岳」とプリポスト環境という異なるシステムに対して，同じURLのWebフォームを利用するため、Dynamic Form Widgetsの機能を用いる（右図）

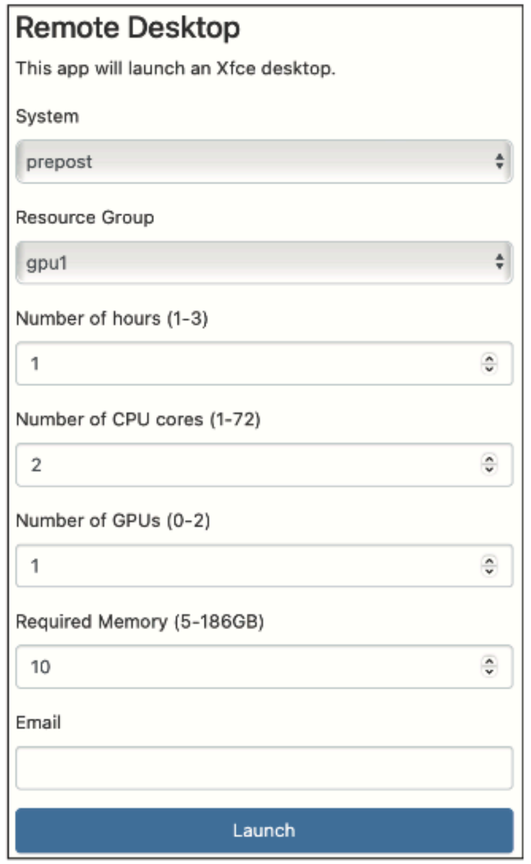
System	Resource group	Time (hours)	CPU Cores	Memory (GB)	GPUs
Fugaku	small-s3	72	-	-	-
Prepost	gpu1	3	72	186	2
	gpu2	24	36	93	2
	mem1	3	224	6,045	-
	mem2	24	56	1,511	-

「富岳」



「システム」の内容により設定項目が変化する。ただし、バグのため、最大値の設定などは正常に動作しない場合がある（報告済）

プリポスト環境



まとめと今後の課題

- まとめ
  - HPCクラスタが持つ計算資源を簡易に利用可能にするOpen OnDemandを「富岳」に導入
  - 「富岳」のジョブスケジューラであるFujitsu TCSを利用するためのアダプタの開発
  - 「富岳」とプリポスト環境という2つのシステムに対するOpen OnDemandの設定
- 今後の課題
  - Open OnDemandを実サービスとして提供し、その導入の効果を定量的に示す
  - 「富岳」のアカウント数は2,822（2022年7月時点）であるため、Open OnDemandの利用者が多くなった場合、複数台のOpen OnDemandサーバを用いた負荷分散の工夫が必要
  - Open OnDemandを用いると、様々なHPCクラスタに共通のフロントエンドでシステムを構築できるため、システムの開発コストの低減とユーザに対する統一されたインタフェースの提供が可能になる。そのため，「富岳」におけるOpen OnDemandの経験を今後も公開していくことで、Open OnDemandの普及を促進したいと考えている