

Web 型プログラミング教育システムの一般公開に向けた改修

齊藤 智也¹⁾, 大江 和一²⁾, 岡田 耕一¹⁾, 爲末 隆弘¹⁾, 王 躍²⁾,
筒井 優子²⁾, 丹生 智也³⁾, 竹房 あつ子^{2),4)}

1) 山口大学

2) 国立情報学研究所

3) 国立遺伝学研究所

4) 総合研究大学院大学

info-cc@ml.cc.yamaguchi-u.ac.jp

Renovation of Web-based Programming Educational System for Public Release

Tomoya Saito¹⁾, Kazuichi Oe²⁾, Koichi Okada¹⁾, Takahiro Tamesue¹⁾,
Yue Wang¹⁾, Yuko Tsutsui²⁾, Tomoya Tanjo³⁾, Atsuko Takefusa^{2),4)}

1) Yamaguchi Univ.

2) National Institute of Informatics

3) National Institute of Genetics

4) The Graduate University for Advanced Studies, SOKENDAI

概要

山口大学では、全学部におけるデータサイエンス関連科目の標準的な演習環境とするため、Moodle と JupyterHub を組み合わせた Web 型プログラミング教育システムを構築している。いくつかの大学から本システムの一般公開に関する要望が寄せられているが、現行システムは山口大学の Moodle システム、ファイルサーバ、及び LDAP サーバの仕様を前提として設計・構築されているため、環境に依存する実装箇所が多く、一般公開が困難である。また、現行システムはオンプレミス環境に整備されているが、オンプレミス環境のみでは利用科目数及び同時両者数の増減への対応が困難である。そこで我々は、Web 型プログラミング教育システムを国立情報学研究所が提供している学認クラウドオンデマンド構築サービス (OCS) 上で実行可能なアプリケーションとして提供するための調査・開発を進めている。本稿では、Web 型プログラミング教育システムの概要、環境依存箇所をはじめとする一般公開のための課題、及びこれまでの取り組みについて報告する。

1 はじめに

近年、大学等における情報科学の関連科目を中心として、Jupyter Notebook 及び JupyterHub [1] の活用が広まりつつある [2]。特に、Moodle [3] 等の学習管理システムと JupyterHub を連携させたプログラミング教育環境の構築が進められている [4–6]。また、JupyterHub の拡張機能である Nbgrader [7, 8] が持つ課題の配布・回収・採点機能等は、多人数を対象としたプログラミング教育において有用である。

山口大学では、全学部におけるデータサイエンス関連科目の標準的な演習環境とするため、Moodle と JupyterHub+Nbgrader を組み合わせた Web 型プログラミング教育システムを構築している [9]。

従来の JupyterHub では、科目ごとの受講生一覧や共有フォルダ等の環境設定を反映させながら利用する場合、科目ごとに設定ファイルを入れ替え、システムを再起動しなければならない。複数科目による同時利用や、受講生による授業時間外の利用が困難であるため、学科ごとや科目ごとに個別の JupyterHub サーバを運用する必要がある。多数の科目が JupyterHub を活用する場合、このような仕組みはシステム管理者及び授業担当者の大きな負担となる。

本システムでは Moodle 及び LDAP サーバの情報を活用することにより、システム管理者もしくは授業担当者が特別な操作を行うことなく、多数の科目が自由に、かつ同時に JupyterHub 及び Nbgrader を利用可能である。しかし、現行システムは山口大学の

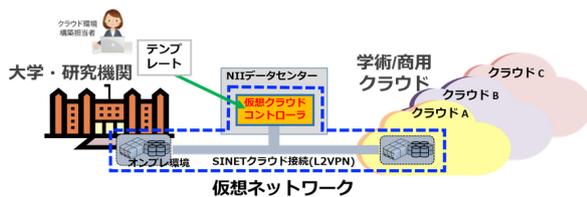


図 1: 学認オンデマンドクラウド構築サービス

Moodle システム、ファイルサーバ、及び LDAP サーバの仕様を前提として設計・構築されている。いくつかの大学より、現行システムの試行についての要望が寄せられているが、現行システムの一般公開は困難である。特に、LDAP サーバで活用されている属性情報等の仕様は大学ごとの相違が顕著である。

また、現行システムでは、利用者ごとの Jupyter 環境は Docker コンテナとして実装されている。多数の科目が同時に JupyterHub を活用するためには、多数の Docker コンテナを同時に稼働させる必要があり、多くの計算機リソースを必要とする。今後、JupyterHub を活用する授業科目、並びに同時利用者数が増加した場合、オンプレミス環境だけで十分な計算機リソースを整備することは困難である。

そこで我々は、山口大学の Web 型プログラミング教育システムを、国立情報学研究所 (NII) が提供している学認クラウドオンデマンド構築サービス (OCS) [10] 上で実行可能なアプリケーションとして提供するための調査・開発を進めている。OCS は、オンプレミス環境とクラウド環境にわたってアプリケーション (Docker コンテナに相当) の配置・実行が可能なクラウド構築支援サービスである。OCS は複数のクラウド・プロバイダに対応しているほか、オンプレミス環境とクラウド環境の一方だけでも利用可能であり、大学の計算機環境及びネットワーク環境に応じた活用が可能である (図 1 参照)。また、OCS では HPC 計算クラスタやオンライン講義演習環境など、研究・教育用途のアプリケーション環境の構築、運用手順を Jupyter Notebook 形式でテンプレートとして公開し [11]、ノウハウの共有を行っている。本研究では、山口大学の JupyterHub+Nbgrader を OCS 上で稼働するアプリケーション・テンプレートとして公開することにより、同システムの他大学への導入が可能になる。

2 現行システムの概要及び問題点

2.1 システムの概要

図 2 に、山口大学における Web 型プログラミング教育システムの概要を示す。利用者は Moodle コース

に設置された外部ツールのインスタンスを經由して、JupyterHub にアクセスする。JupyterHub では LTI (Learning Tools Interoperability) 認証を採用しているため、利用者のログインは自動的に完了する。

JupyterHub へのログインが完了すると、利用者が經由してきた Moodle コース、及びコースにおけるロール (役割) に応じて、自動的に環境設定等が行われる。まず、Nbgrader を利用しない科目向けに、科目ごとの共有フォルダが作成される。共有フォルダのアクセス権限は、Moodle コースの教師は読み書き可能に、学生は読み込み限定に設定される。また、各利用者のホームディレクトリの中に、共有フォルダへのシンボリックリンクが作成される。シンボリックリンクの名称は科目によらず単一であり、リンク先となる共有フォルダは Moodle コースに応じて設定される。

次に、Nbgrader を扱うための環境設定が自動的に行われる。利用者が經由してきた Moodle コースに応じて、課題の配布・提出、及びフィードバックに利用される各フォルダは自動的に作成され、アクセス権限も自動的に設定される。一般に、利用者は Nbgrader コースを選択した後に各種操作を行うが、利用科目数が増えると選択を間違える恐れがある。これに対し、本システムでは、Moodle コースに対応した Nbgrader コースが自動的に選択される。

JupyterHub のターミナルの利用権限、共有フォルダへのアクセス権限を設定するために、LTI 連携により Moodle から取得される情報、及び大学の LDAP サーバから取得される属性情報を利用する。また、利用者が教師として Nbgrader を扱う場合、Moodle のデータベースから受講生一覧を取得する。

2.2 利用状況

2022 年度は、本システムは 13 科目で活用され、最大同時利用者数は 78 名であった。山口大学における JupyterHub の整備目的は、文系学部を含む全学部のデータサイエンス関連科目における標準的な演習環境とするためである。現在は全学部において「データサイエンス技術」等の専門教育科目が設置されている。これらすべての科目で本システムが利用される場合、年度ごとの利用科目数は 30 科目以上、最大同時利用者数は 300 名と想定されている。これまでの利用状況が当初の想定に到達していない原因は、利用者ごとの Jupyter 環境を起動するための実行サーバの整備が遅れているためである。

JupyterHub の標準設定では、利用者ごとの Jupyter 環境や、Jupyter 環境から実行されるプログラムは、

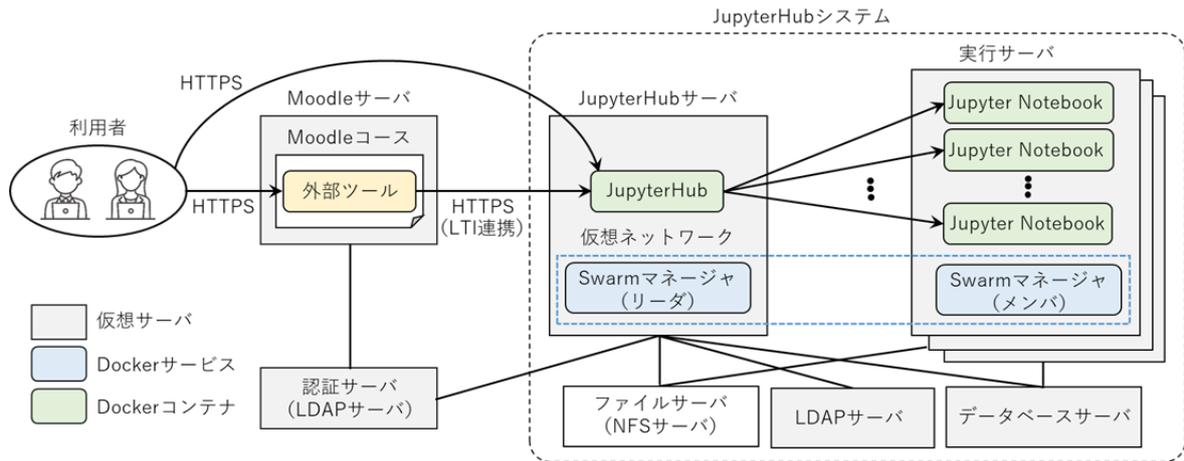


図 2: JupyterHub を用いた Web 型プログラミング教育システム

JupyterHub サーバの OS 上のプロセスとして起動される。利用者ごとのプログラムの実行環境が JupyterHub のそれと分離されていないため、セキュリティ面の問題が生じやすい。また、複数のサーバによる負荷分散が困難である。このような動作形態は全学部で自由に利用可能な演習環境として不適切であるため、本システムでは、Docker Swarm を用いた動的負荷分散を採用している。利用者の Jupyter 環境は、Docker Swarm のノード（実行サーバ）上に Docker コンテナとして起動される。

利用者の Jupyter 環境を Docker コンテナとして起動する場合、各コンテナに割り当てる CPU コア数は、JupyterHub の設定ファイルに固定値として記述される。そのため、実行サーバの総 CPU コア数によってシステムの収容可能人数が制限される。現時点では、山口大学の JupyterHub の収容可能人数は 120 名であるため、受講生が 100 名以上になる科目/クラスについては、システムの利用を控えている。

実行サーバを増強して CPU コア数を増加させることにより、システムの収容可能人数を 300 名まで増加させることは可能である。しかしながら、サーバ室の占有スペースや消費電力の面から、必要なサーバ設備を長年にわたってオンプレミス環境のみで維持し続けることは困難である。一方、受講生が多いクラスの演習が実施される際には、大量の CPU コア及びメモリが利用されるため、クラウド環境のみによる稼働は費用の面で困難である。このような課題に対処するため、OCS を活用することとした。

2.3 一般公開に対する問題点

現行システムが山口大学の計算機環境及び Moodle システムの仕様に依存している理由として、以下の点

が挙げられる。

- 組織ごとに LDAP サーバにおいて使用されている属性項目及び属性値に差異がある。
- JupyterHub は Moodle の DB サーバに接続し、SQL クエリを発行して履修登録者の一覧を取得しているが、DB サーバに直接に接続する仕組みは開発元の環境に依存しやすい。
- JupyterHub の設定ファイルに、山口大学の計算機環境及び Moodle に依存した設定値が直接に記述されている。

また、Moodle と JupyterHub の間の LTI 認証に LTI 1.1 を採用している点や、JupyterHub のバージョンが 1.4 である点など、採用しているソフトウェア及び標準化技術が既に古いものである。アプリケーション・テンプレートの一般公開に際し、これらのバージョンアップに関する検討も必要である。

3 これまでの取り組み

Web 型プログラミング教育システムの一般公開のため、OCS 上のアプリケーションとしてシステムを実装することとした。OCS 用のアプリケーション・テンプレートとしてシステムを一般公開することにより、他の大学等への導入が容易になる。

2023 年 9 月時点では、現行システムと同様の機能を有する JupyterHub+Nbgrader をアプリケーション・テンプレートとして実装し、NII 側の OCS 開発環境で動作確認を行うところまで進めた。また、アプリケーション・テンプレートの作成にあたり、JupyterHub の設定ファイル (jupyterhub_config.py) を整理した。設定値が直接に記述されていた箇所を修正し、組織ご

とに設定値が異なる箇所については、それらの値をテンプレート内でパラメータとして指定・記述できるよう、設定ファイルから分離した。

2.3 節で示した 3 点の課題のうち、今後は認証サーバ (LDAP) の取り扱い、及び履修登録者の一覧の取得について調査・開発を進める予定である。

4 まとめ

本稿では、Moodle と JupyterHub+Nbgrader を組み合わせた Web 型プログラミング教育システムの一般公開のための修正・改善について述べた。現行システムでは山口大学の計算機環境、及びネットワーク環境に依存する箇所が多く、他の組織への導入が困難である。また、現行システムはオンプレミス環境での Docker Swarm を用いて稼働するように設計されているため、利用科目数及び同時利用者数の増加に対応するためのサーバ設備を確保し続けるが困難である。

そこで我々は、Web 型プログラミング教育システムを OCS 上のアプリケーションとして稼働するよう修正することとした。ハイブリッド・クラウド環境への移行により、組織ごとの計算機環境に応じた実装・運用が容易になる。加えて、アプリケーション・テンプレート的一般公開により、システムの他大学への導入を容易にすることを目指している。

これまでに、現行システムと同様の機能を有する JupyterHub+Nbgrader をアプリケーション・テンプレートとして実装し、NII 側の OCS 開発環境で動作確認を行うところまで進めた。また、JupyterHub の設定ファイルの修正を進めた。システムの運用組織ごとに設定値が異なる箇所については、それらの値をテンプレート内でパラメータとして指定・記述できるよう、設定ファイルから分離した。

今後は山口大学側に OCS 実行環境を準備し、クラウド環境、オンプレミス環境、及びハイブリッド・クラウド環境における動作確認を順次進める予定である。

また、今後の課題として、2.3 節で述べた 3 点の課題のうち、認証サーバ (LDAP) の取り扱い、及び Moodle コースの受講生一覧の取得について、山口大学の環境に依存する箇所を修正することが挙げられる。

謝辞

この研究は 2023 年度国立情報学研究所公募型共同研究 (23S1003) の助成を受けています。

本研究にご協力いただいた株式会社アスケイドの那須野淳様、羽鳥文子様、増山隆様、神田千暁様、鳥井

貴史様、寺山流晟様に深く感謝いたします。

参考文献

- [1] JupyterHub, <https://jupyter.org/hub> (2023 年 9 月 25 日参照).
- [2] 山口健二, 「Google Colaboratory による自宅学習教材の作成と遠隔授業方法の検討」, 高等教育と学生支援: お茶の水女子大学紀要, vol.10, pp.59-63 (2019).
- [3] Moodle: Open-source learning platform, <https://moodle.org/> (2022 年 10 月 5 日確認).
- [4] 井関文一 他, 「LTI カスタムパラメータによる Moodle - JupyterHub 連携に関する研究」, 日本ムードル協会全国大会 (2022) 発表論文集, pp.20-25 (2022).
- [5] 池田裕希 他, 「Moodle と JupyterHub を用いたプログラミング環境の構築」, 情報処理学会第 84 回全国大会 (2022) .
- [6] 石坂徹 他, 「Moodle と Jupyter Notebook の連携によるプログラミング教育環境の構築」, 日本ムードル協会全国大会 (2019) 発表論文集, pp.32-37 (2019).
- [7] Nbgrader, <https://github.com/jupyter/nbgrader> (2023 年 9 月 25 日確認).
- [8] Nik Klever, 「Jupyter Notebook, JupyterHub and Nbgrader」, https://klever.hs-augsburg.de/nb/OWL/LAB_2020_02_Klever_Jupyter_Notebook_JupyterHub_Nbgrader.pdf (2023 年 9 月 25 日確認).
- [9] 齊藤智也 他, Moodle と JupyterHub を用いた Web 型プログラミング教育環境の改善および運用, 情報処理学会研究報告, 2023-CLE-39(8) (2023).
- [10] 学認クラウドオンデマンド構築サービス, <https://cloud.gakunin.jp/ocs/> (2023 年 9 月 25 日確認).
- [11] 学認クラウドオンデマンド構築サービスのアプリケーションテンプレート, <https://github.com/nii-gakunin-cloud/ocs-templates> (2023 年 10 月 4 日確認).