Vol.2019-CLE-27 No.20
2019/3/22

Notebookによる講義・演習環境の開発

長久勝[†1]　政谷好伸[†1]　合田憲人[†1]

概要：Jupyter 形式の Notebook は、国内外の大学で、データサイエンスやプログラミングの演習教材として広く使われている。また講義・演習に際しては、多数の受講生がコンテナ化された Notebook 演習環境(コンテナ化された Jupyter サーバ）を Web から利用できる「JupyterHub」という仕組みが利用できる。筆者らは Notebook や JupyterHub を活用したシステムの構築と運用を実践しているところであるが、今回は、その中で養った知見を元に JupyterHub を拡張した講義・演習環境「CoursewareHub」を開発したので、その概要について述べる。CoursewareHub では独自に開発した Notebook 用のプラグインの適用や、一つの JupyterHub 環境の中に講師用と受講生用と言った役割の異なるコンテナを配備できるようにするなどの工夫を施し、より使いやすい演習環境の提供を目指している。例えば、(1)Notebook 教材や演習を実行するための環境を講師が CoursewareHub 上でワンストップで開発できる、(2)講義に際して教材の配布、中途での教材更新・再配布、演習課題の回答回収などが簡便に実施できる、(3)受講生の Notebook 操作履歴を取得し、学習活動の分析に利用することができる、(4)学認 mAP を活用した利用の登録・管理ができる。本稿では、CoursewareHub の特徴と、試用結果や評価について述べる。

CoursewareHub:
A classroom tool for conducting lessons based on Notebooks

MASARU NAGAKU[†1]　YOSHINOBU MASATANI[†1]　KENTO AIDA[†1]

1. はじめに

Jupyter[1]形式の Notebook は、国内外の大学で、データサイエンスやプログラミングの演習教材として広く使われている。例えば、Jupyter 関連プロダクトで主導的な役割を果たす Project Jupyter が英語圏におけるさまざまな取り組みをまとめた「A gallery of interesting Jupyter Notebooks」[2]や、国内の教材で公開されているものとして京都大学工学部の情報基礎演習[3]や京都大学農学部のゲノム博物学入門[4]がある。少し検索すれば、いくらでも見つかるような状況である。

Notebook を用いた講義・演習に際しては、多数の受講生が、コンテナ化された Notebook 演習環境(コンテナ化された Jupyter サーバ）を、web ブラウザから利用できる「JupyterHub」という仕組みが利用できる。Project Jupyter をホストする UC Berkeley では、学内インフラとして JupyterHub を整備しており[5]、UCLA[6]などにも同様の動きが広がっている。

筆者らは、Notebook や JupyterHub を活用したシステムの構築と運用の手法「Literate Computing for Reproducible Infrastructure」[7][8]（以下、LC4RI）を、OpenStack によるクラウド基盤の構築・運用において実践しており[9]、この活動を通して、Notebook 活用や Jupyter のプラグイン拡張といった知見を有している。

また、国立情報学研究所の社会人向け教育プログラム「トップエスイー」[10]におけるクラウド関連の講義・演習環境準備にも関わっており、Notebook 教材を用いた演習環境の構築支援を行って来た。そこでの経験において、計算機を活用した講義・演習を運営する上での課題に気づいたり、課題を軽減・解消するための工夫を行ったりする中で、JupyterHub を拡張した講義・演習環境「CoursewareHub」[11]を開発するに至った。現在 CoursewareHub は、その評価を目的に、群馬大学、室蘭工業大学の研究グループに提供している。また、評価のベースになっているコード類は GitHub 上に公開している。

本稿では、CoursewareHub に関する一連の取り組みについて述べる。

2. Jupyter Notebook を用いた演習における課題

Jupyter Notebook を、個々人が研究環境として構築し利用するためのリソースは良く整備されている[12]。しかし、複数の受講生に対して演習環境として提供する場合、アカウントの管理や、実行環境の分離など、いくつかの機能が不足している。

これに対する答えとして、Project Jupyter は、JupyterHub[13]を開発した。複数の受講生に対して Web ベースで利用可能な演習環境を、サーバとして提供する JupyterHub は、先に引用した通り、UC Berkeley のコロキウムでの大規模な演習等ですでに活用されている。

JupyterHub には複数のアカウントを管理するとともに、アカウント毎に分離した実行環境を多数並列に提供するなどの機能がある。しかし、JupyterHub は、演習のライフサイクル全体をカバーするには至っていない。

- 演習内容に合わせた実行環境の作り込みを含む教材の準備

†1 国立情報学研究所
National Institute of Informatics

ⓒ 2019 Information Processing Society of Japan

- 教材の配布・更新
- 課題の回収
- 学習活動の分析

などの機能が備わっていない。このため、実行環境の作り込みにエンジニアの支援が必要となったり、GitHub など外部のリポジトリを通じて教材を配布する必要があったり、教材の軽微な誤りを見つけて修正しても迅速に再配布する手段がなかったり、いくつかの場面で不便がある。

3. CoursewareHub のアプローチ

規模がスケールするプラットフォームを提供することを目的としている JupyterHub に対して、筆者らは、講義・演習の準備から実施までに渡る煩雑な講師の作業負担を軽減することを目的として以下のような機能追加、拡張を実装した CoursewareHub を開発した。

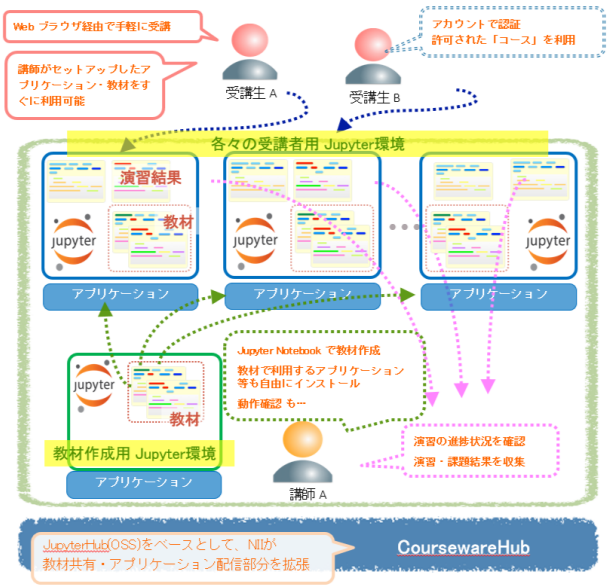


図 1：CoursewareHub の概要

3.1 JupyterHub 上へ講師用コンテナを配備する機能

通常の JupyterHub には、受講生用 Jupyter Notebook 実行環境をコンテナとして演習時に多数配備する機能があるが、演習用の教材は講師が独自の環境で開発する必要がある。また、教材のテスト等は実際の演習環境である JupyterHub 上で行う必要がある。CoursewareHub ではひとつの JupyterHub の中に、講師用の開発環境と受講生用の演習環境を並立させている。演習時に受講生が Web ブラウザ経由で受講できるだけではなく、講師も Web ブラウザ経由で Notebook 教材をサーバ/クラウド上で作成することができる。また、講師用開発環境のコンテナイメージから直接、受講生用の演習コンテナイメージを生成するため、開発環境と演習環境の差異がない。講師はコンテンツの作成に専念できる。

3.2 受講生のアカウントを管理する機能 -CSV による一括登録と学認 mAP の利用

CoursewareHub では利用者のアカウントを CSV ファイルなどを用いて一括登録･管理する方法の他に、より利便性の高い「学認 mAP」[14]を活用した認証連携を可能としている。後者の方法では、受講生は学認の SSO（Single Sign-On）を用いて認証され、CoursewareHub 上の講義･演習を履修することができる。CoursewareHub では学認 mAP を利用した際に送信されてくるグループ属性値を元に、講義・演習の利用を認可するようにしている。そこでは、(a) 講義・演習毎に CoursewareHub のクラスタを用意する、(b) CoursewareHub のインスタンス毎に受け入れるグループ属性（複数指定可能）を設定する、(b) 講義・演習の受講生は学認 mAP を用いて管理するといった運用形態を想定している。プラットフォームの運営・リソース管理主体（作業 a, b）と受講生を管理する講義・演習の主催（作業 c）の責任分解点を明確にするとともに、システム内に受講生のクレデンシャルを直接保持する必要のない運用が可能な設計となっている。

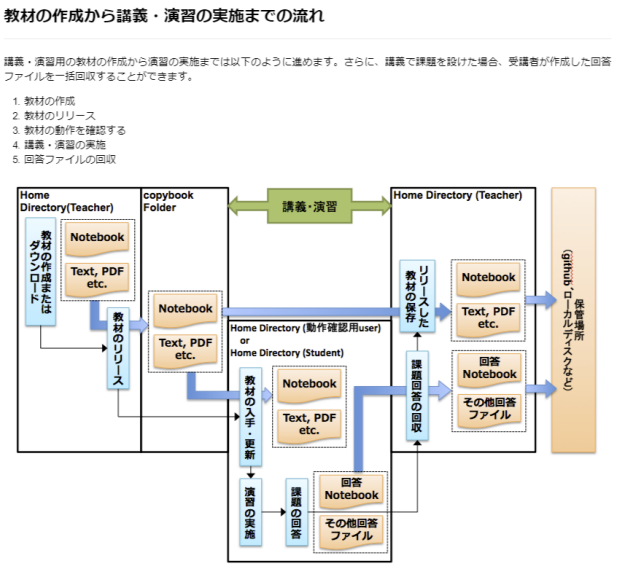


図２：教材の作成から講義・演習の実施までの流れ

3.3 講師のワークフローをサポートする機能 -教材配布、講義・演習途中の更新、演習の回答を回収する

通常の JupyterHub を用いた演習では GitHub などのリポジトリに教材を配置し、各受講生が自ら JupyterHub 環境に取り込んで演習を実施する。いったん GitHub から git clone したり、ローカルの環境経由で教材をアップロードするなどの受講生の作業が必要となる場合が多く、受講生には一定のリテラシが求められる。また、講師は予め教材を用意し事前にリポジトリ登録しておくことが求められる。さらに、演習で独自のアプリケーションを用いる場合は、そのアプリケーションを組み込みカスタマイズしたコンテナイ

ⓒ 2019 Information Processing Society of Japan

メージを講師が予め作成し（Docker イメージのビルドノウハウが必要）、 JupyterHub の運用担当者にイメージの登録を依頼する、あるいは、演習時間の一部を割いて各受講生にアプリケーションのダウンロードとインストールを実施させるなどの付加的な作業が必要にもなる。

CoursewareHub では、先に説明した講師用の開発環境コンテナから、講師が自ら作成した教材を、自ら各受講生のコンテナに配信する機能がある。演習の途中で教材を更新・追加配信することも容易に実施できる。また、演習中の必要なタイミングで、受講生の回答を回収することができる。さらに、演習で独自のアプリケーションを利用したい場合、講師が開発環境で一般的な方法でインストール作業を行えば（pip, anaconda, yum などの利用が可能）、カスタマイズした環境からスナップショットでコンテナイメージを作成し、各受講生の利用するコンテナとして CoursewareHub に自動登録することができる。

3.4 受講生の演習実施履歴を収集する機能

CoursewareHub の Jupyter Notebook 実行環境には、筆者らが開発した LC_nblineage[15]と LC_wrapper[16]を組み込んでいる。LC_nblineage には Notebook 内のひとつひとつの Cell にユニークな識別子を付与する機能がある。LC_wrapper は Cell が実行される度に、当該のユニークな識別子、実行時刻、実行内容、実行結果などを、都度々々外部ファイルに書き出す機能がある。これらにより、受講生が課題の Cell 毎に、どのような時刻、順序で試行錯誤しながら実行したといった履歴、さらには課題の設定から外れ独自に Cell を追加するなどして試行錯誤した履歴も齟齬なく収集することができる。CoursewareHub では全受講生の履歴を一元的に収集することが可能である。

収集した実施履歴を用いることで、例えば、課題として配布した Notebook に対して、各受講生がどこまで試行したかの経緯を観察することにより、受講生の進捗度合いを演習実施時に把握する、課題毎に要した時間を集計することにより課題の難易度を評価するなどの、分析が可能になると考えている。

4. CoursewareHub の評価

ここでは、トップエスイーの分散アプリ演習の一部で CoursewareHub を実利用したケースと、室蘭工業大学での学習活動の分析について評価を行ったケースについて述べる。

2018 年度のトップエスイーの分散アプリ演習では、Apache Spark[17]を使ったプログラミング演習の後で、Spark 環境を Docker で構築し開発環境として利用できることを示す演習を行った。この際、LC4RI の手法を紹介する目的もあり、CoursewareHub を演習環境とした。教材の配布と回収は問題なく機能し、回収した Notebook を確認して成績をつけた。トップエスイーでは、講義期間中のみ演習環境を維持することで、無駄なコンピュートリソースを持たないようにしているが、これまでは、受講生の学習活動が演習環境上にしか残らない等の理由で、迅速にリソースを開放できることが少なかった。CoursewareHub を用いることで、回収した Notebook に受講生の学習活動が保存され、演習に使ったコンピュートリソースを迅速に開放できた。回収した Notebook では、課題を完成できなかった受講生が多く見られたが、そこに至る例題は実行されており、課題の難易度に問題があった可能性を見つけることができた。

室蘭工業大学のグループ[18]では、来年度から Notebook によるプログラミング演習を予定しており、CoursewareHub が持つ受講生の演習実施履歴を収集する機能について、評価を行っている。Cell 毎の実行回数や実行時刻によって、何度もやり直した箇所や、Notebook を読む速さなどが分かるため、理解度を測る補助的な情報として使ったり、教材の改善につなげることが、可能ではないかと期待できる。

5. CoursewareHub の発展

CoursewareHub について、いくつかの研究・教育機関と議論する中で、さまざまなユースケースが得られたため、それらへの対応を今後の課題として認識している。ここでは、そのうちの 2 つについて述べる。

1 つめは、LMS（Learning Management System）との LTI（Learning Tools. Interoperability）連携について述べる。CoursewareHub は演習環境として使われることを想定しているが、近年の講義運営では LMS を利用することが一般的になっており、CoursewareHub と LMS をどう連携させるかに課題がある。1 つの解として、学認 mAP を使って、CoursewareHub と LMS で SSO による透過的な利用ができる仕組みを開発した。しかし、これとは別に、受講生のグループ管理を、LMS 上の情報に基づいて行えるようにしたいというユースケースがあった。現在、よく使われている LMS の多くは、LMS と他のシステムを連携させるための標準である LTI（Learning Tools. Interoperability）対応であり、CoursewareHub と LMS で LTI による連携を期待する機関が多い。室蘭工業大学のグループ[19]では、Moodle と JupyterHub の LTI 連携について技術検証を行っており、これを参考に、CoursewareHub と Sakai などの組み合わせでの LTI 連携についても、検証を行う予定である。

2 つめは、GPU を使える演習環境について述べる。Jupyter 形式の Notebook を使った演習では、データ分析や機械学習などを扱うために、NVIDIA 社の GPU に依存したライブラリが使われることが多い。これに対応するためには、Notebook の実行環境において、GPU ドライバにリクエストできる状態が要求される。CoursewareHub では、Notebook の実行環境として、Docker コンテナがスポーンされるので、Docker エンジンを nvidia-docker[20]化することで、Notebook 上で NVIDIA 社の GPU に依存したライブラリが使えるよ

ⓒ 2019 Information Processing Society of Japan

うになる。CoursewareHub の配備手順として、nvidia-docker 化を取り込んだものを用意し、高い需要の見込まれる領域での活用が可能となるよう、取り組みを進めて行く予定である。

6. CoursewareHub と LC4RI

3 章では利用者の視点から CoursewareHub の特徴について述べた。ここでは実装上の工夫について説明する。筆者らは、Notebook や JupyterHub を活用したシステムの構築と運用の手法 LC4RI を実践している。LC4RI では構築や運用の手順を Notebook として記述する人間中心の機械化を追求している。Notebook で書かれた実行可能な手順は、機械的に実行することも、必要に応じて人間が介在することもできる。また、ドキュメントとして手順と実行結果・証跡が齟齬なく一体となっているので、作業の前提条件など必要に応じて読み解く余地が残されている。このような手順の機械的な実行と、実行内容の見える化を両立させることのできる道具立てを用いることで、機械化（Automated Operation）を追求して自動実行の守備範囲を拡げ、高度化するといった継続的なプロセスにおいて、いわゆる「オートメーション・バカ」[21]の罠に過度に突き進むことを避けることができる。

CoursewareHub においては、様々なワークフロー機能を実装するにあたって LC4RI の手法を踏襲している。具体的には講師のワークフローを支援する機能は Notebook として実装している。ワークフローの実装を Notebook として見える化することによって、CoursewareHub の振る舞いを講師が自らカスタマイズすることができる。図 3 に講師用のワークフローの実装例を示す。

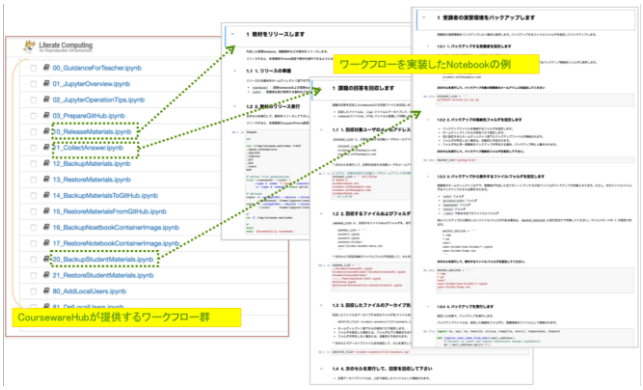


図 3 ：ワークフローの例

7. おわりに

本稿では、筆者らが LC4RI を通じて得た Jupyter Notebook のノウハウを転用して、JupyterHub に足りない機能を補い開発した CoursewareHub について報告した。

JupyterHub を用いた演習は、多くの教育機関が取り組みはじめており、今後、利用が拡大していくことが予想されるが、我々の事例を参考に、より良い演習環境を作り上げてもらえれば幸いである。本稿で取り上げた筆者らの開発成果物は、GitHub で公開されているので、不明な点などあれば、問い合わせていただきたい。

今後も、さまざまな教育機関と連携し、現場のユースケースに合う形で CoursewareHub を導入いただけるように、取り組みを進めたい。

参考文献

[1] ”Project Jupyter”. https://jupyter.org/ , (参照 2019-02-26).
[2] "A gallery of interesting Jupyter Notebooks". https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks , (参照 2019-02-26).
[3]「京都大学工学部 情報基礎演習」. https://github.com/Johokiso-enshu/textbook , (参照 2019-02-26).
[4]「ゲノム博物学入門」. https://github.com/CropEvol/lecture , (参照 2019-02-26).
[5] ”The course of the future – and the technology behind it”. https://data.berkeley.edu/news/coursefuture , (参照 2019-02-26).
[6] "UCLA IDRE JupyterHub". http://jupyter.idre.ucla.edu/ , (参照 2019-02-26).
[7] Masatani, Y.. Collaboration and automated operation as literate computing for reproducible infrastructure. JupyterCon, 2017-08-25.
[8] 長久勝, 政谷好伸, 谷沢智史, 中川晋吾, 合田憲人. Literate Computing for Reproducible Infrastructure による研究・教育環境の構築と運用. 大学 ICT 推進協議会 2017 年度年次大会, 2017-12-13.
[9] 長久勝, 政谷好伸, 横山重俊, 谷沢智史, 中川晋吾, 合田憲人. Academic Baremetal Cloud の実現. 大学 ICT 推進協議会 2018 年度年次大会, 2018-11-19.
[10]「トップエスイー」. https://www.topse.jp/ja/ , (参照 2019-02-26).
[11] "CoursewareHub". https://github.com/NII-cloud-operation/CoursewareHub-LC_platform , (参照 2019-02-26).
[12] "Teaching and Learning with Jupyter". https://jupyter4edu.github.io/jupyter-edu-book/index.html#acknowledgments , (参照 2019-02-26).
[13] "JupyterHub". https://jupyter.org/hub , (参照 2019-02-26).
[14]「2011 年度 SINET&学認説明会 学認最新 R&D 報告」. https://www.sinet.ad.jp/wp-content/uploads/2018/11/4-gakunin-R-D.pdf , (参照 2019-02-26).
[15] “Jupyter-LC_nblineage”. https://github.com/NII-cloud-operation/Jupyter-LC_nblineage , (参照 2019-02-26).
[16] “Jupyter-LC_wrapper”. https://github.com/NII-cloud-operation/Jupyter-LC_wrapper , (参照 2019-02-26).
[17] "Apache Spark". https://spark.apache.org/ , (参照 2019-02-26).
[18] 桑田喜隆, 石坂徹, 小川祐紀雄, 政谷好伸, 長久勝, 横山重俊, 浜元信州. Jupyter Notebook の実行履歴を活用したプログラミング演習の状況把握. 人工知能学会 知識流通ネットワーク研究会 第 24 回研究会, 2019-03-08.
[19] 石坂徹, 桑田喜隆, 合田憲人, 政谷好伸, 横山重俊. Moodle と Jupyter Notebook の連携. MoodleMoot Japan 2019, 2019-03-01.
[20] "nvidia-docker". https://github.com/NVIDIA/nvidia-docker , (参照 2019-02-26).
[21] ニコラス・G・カー. オートメーション・バカ-先端技術がわたしたちにしていること. 青土社. 2014.

ⓒ 2019 Information Processing Society of Japan