OpenHPCv2上での機械学習アプリケーション環境の構築

2022年3月25日

佐賀 一繁

国立情報学研究所
クラウド基盤研究開発センター

背景と目的

背景

- 学認クラウドオンデマンド構築サービスでは、利用者の利便性向上のため、アプリケーション環境の構築テンプレートを公開している（まだ、僅かですが… ^^;）
  https://github.com/nii-gakunin-cloud/ocs-templates/

目的

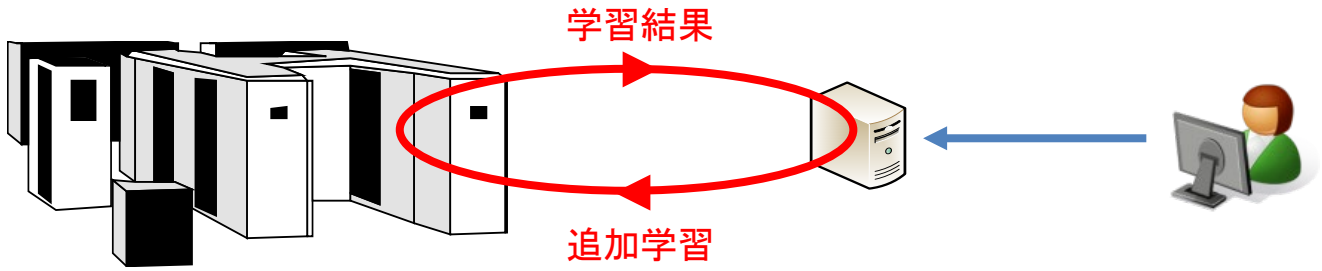
- 単なる OCS の使用法ではなく、公開テンプレートによる実用的な環境の構築をハンズオンで体験していただく

今回のハンズオン

- 公開テンプレートの1つである HPC-v2 テンプレートにより、クラウド上に OpenHPC 環境を構築する
- OpenHPC 環境で機械学習アプリケーションを実行可能とするため、GPU と TensorFlow をサポートする
- OpenHPC 環境で動作する機械学習アプリケーションとして、MNIST データによる手書き数字認識を実行する。この時、単に学習だけでなく総合的な認識システム構築例を示す

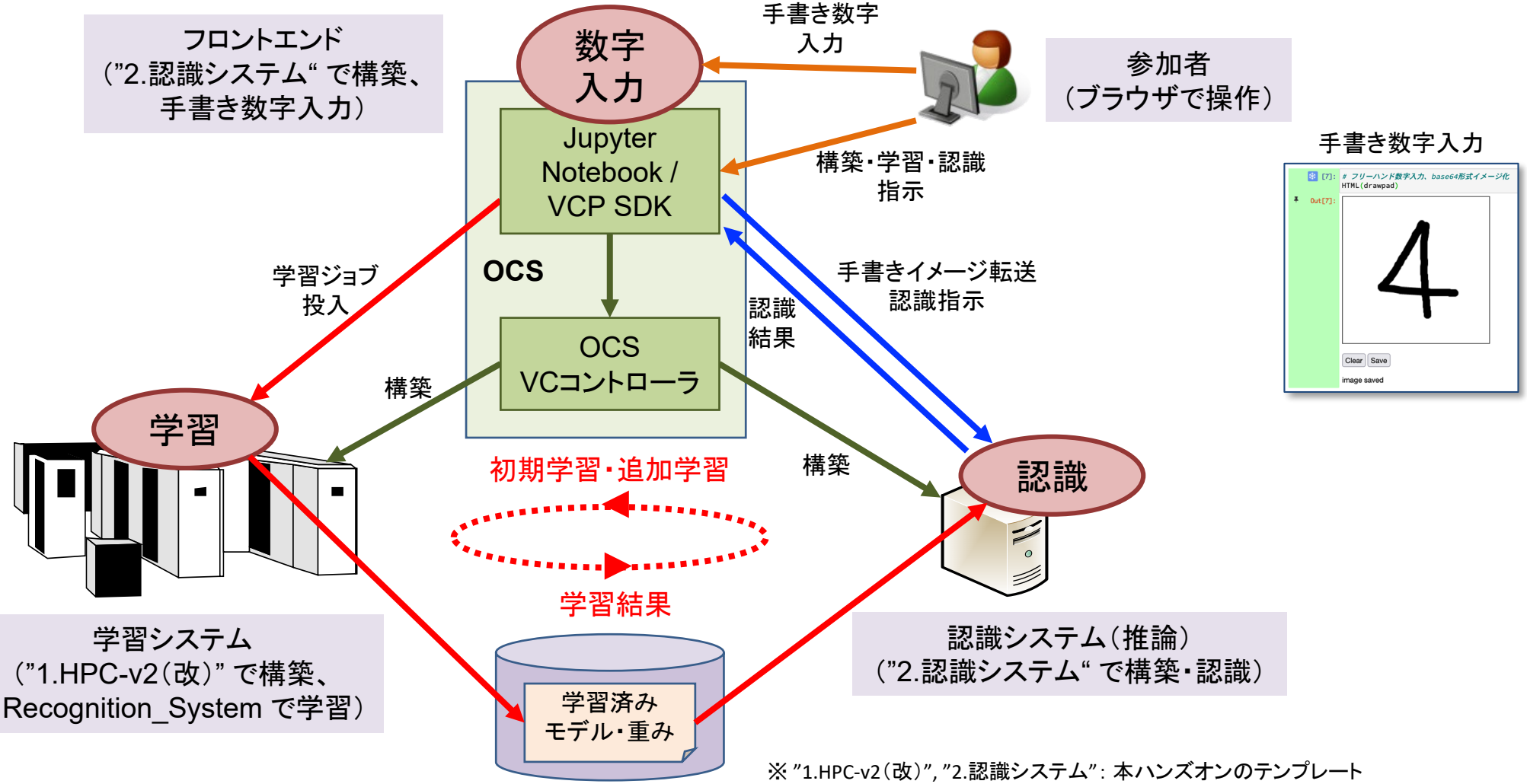
機械学習によるアプリケーション実行環境（一般論）

- 学習とアプリ実行（推論）のシステムを分けることが多い
  - 理由1： 必要な計算性能、メモリ・ストレージ容量、コスト、消費電力…
    - **学習 >> 推論**
    - 例： パターン認識システム（画像認識、文字認識…）
      - 通常はアプリ実行システムで推論（認識）を実行
      - 誤認識が多くなったら、学習システムで追加学習し、アプリシステムに反映
  - 理由2： 学習データの置き場所の問題
    - 学習データは秘密性が高いため、一時的にでもアプリ利用者が利用する環境に置きたくない
      - 大規模な学習には月単位の時間を要する場合もある。追加学習時に問題となる



本ハンズオンで構築する環境

■ 手書き数字認識システム



構築する環境の内容

- 学習システム
  - OpenHPC-v2 による GPU ノード環境、NGC/TensorFlow コンテナが動作
- 認識システム（推論システム）
  - CPU のみの環境、TensorFlowコンテナが動作
- フロントエンド
  - JupyterNotebook上に下記機能を実装、手書き数字認識が動作
    - 学習ジョブ投入
      学習システムに MNIST の学習ジョブを投入、学習モデルと学習結果としての重みを回収
    - 認識システムへの学習結果展開
      学習モデルと学習結果を認識システムに転送し展開
    - 手書き数字入力認識システム
      手書き数字の入力しイメージファイル化して認識システムに転送、認識指示

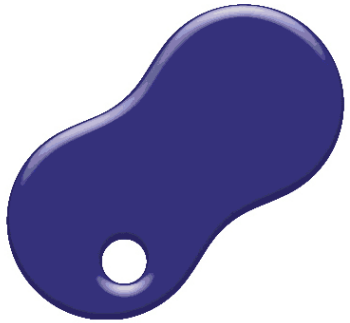
テンプレートとハンズオンの流れ

- **1.HPC-v2（改）**
  - OpenHPC 環境をクラウド上に構築するテンプレート
  - 公開中の HPC v2 テンプレートをハンズオン環境用に微修正したもの
  - 流れ
    - 学習システムの構築
      - プロバイダ：AWS
      - ジョブマネージャ：Slurm
      - ノード: GPU ノード(g3s.xlarge) x 1
      - 機械学習フレームワーク：TensorFlow（NGCコンテナ）

- **2.認識システム**
  - 本ハンズオンのためのテンプレート
  - 流れ
    - 認識システムの構築
      - プロバイダ：AWS
      - ノード: CPU ノード(g3s.xlarge) x 1
      - 機械学習フレームワーク：TensorFlow（コンテナ）
    - 学習システムで MNIST データの学習
    - フロントエンドで手書き数字認識
      - 手書き数字入力の設定
      - 手書き数字入力と認識システムでの認識
      - 認識精度向上、コスト低減のための取り組み

- **両システムの削除**
  - 両テンプレートの削除機能を利用

GakuNin Cloud



大学共同利用機関法人 情報・システム研究機構

国立情報学研究所

National Institute of Informatics