

1 台のミドルパワーPC から始めるスケールアウト可能な地震解析環境構築

岡田 和見

北海道大学大学院理学研究院附属地震火山研究観測センター

1. はじめに

近年、地震学分野では PhaseNet (Zhu & Beroza, 2019) [1]や GaMMA (Zhu et al., 2022) [2]などの機械学習技術の発展により、従来の手法と比較して地震検出数を大幅に増加できるようになった。これらの技術を活用し、地震波形データの前処理から P 波と S 波の到着時刻の検出、震源決定までの一連の処理を自動化・高速化する分散ワークフローとして、QuakeFlow (Zhu et al., 2023) [3]が提案されている。Zhu et al. (2023)では、QuakeFlow の分散計算処理の基盤としてクラウド環境を使用し、地震波形データの処理を高速化できたことが報告されている。

機械学習に基づく自動検出技術の導入を検討する段階で、最初からクラウド環境を使って検証を行うことは環境構築作業及び環境利用の予算の面で高いハードルがある。そこで本報告では、先行研究で提案されている QuakeFlow のワークフローを技術検証対象とし、ローカル環境にある 1 台のミドルパワーPC への導入から始めて、クラウド環境への移行作業まで段階的にスケールアウト（プロセス処理単位を並列に増やす）可能な環境構築手法を検討した。

2. 検証環境・手法

本検証では、ローカル環境として Intel Core i9-10900K + RTX 3070 を搭載した PC、クラウド環境としてさくらのクラウド上の NVIDIA V100 GPU 環境を使用する。検証データには、QuakeFlow の GitHub 公開リポジトリで Demo データセットとして配布されている 2019 年 7 月 4 日にアメリカ合衆国カリフォルニア州 Ridgecrest で発生した地震の震源域周辺の地震の検出（P 波や S 波の到着時刻）データと、防災科学研究所のサイトからダウンロード可能な 2018 年 9 月 6 日に発生した北海道胆振東部地震の震源域周辺の検出データを採用する。

地震波形データ処理のための Python プログラム群を動かす環境構築には、コンテナ技術を利用して仮想環境を実現する Docker を使用し、ローカル環境 PC 上には Docker Desktop Kubernetes を、クラウド環境側にはバーチャルマシン上に構築した Linux サーバー上に軽量なコンテナランタイムソフトウェアの Containerd をインストールし、さらに Kubernetes を導入した。Docker を使用すると、ローカル PC とクラウド環境といった全く違うマシン環境でもその違いを気にせず、プログラムの実行環境を作成して動作を確認することが可能である。また、本検証の処理ワークフローの構築において JupyterLab の拡張ツールである Elyra Pipeline Visual Editor を採用した。これについては後述する。

本報告の検証作業においては、簡易なコマンドライン入力によって高度なコンピュータ操作支援を行うことができる Claude Code と、AI エージェント機能を用いてプログラムコード生成やデバッグ作業を協働できる Cursor を活用し、ユーザー側のクラウド環境や分散計算基盤技術の専門的な知識の不足を補い、同時に技術習得を行いながら実施した。こちらの詳細については、本研究会二日目の情報系交流企画ワークショップにて報告する。

3. Elyra による機械学習プロセスのコンポーネント化

Elyra (エライラ) は、JupyterLab の拡張機能である。Elyra を使うと、「データの読み込みや前処理」、「機械学習モデルを用いた地震波形データ処理」、「処理されたデータを用いた推論」、「その結果のまとめ」といった一連の作業の流れを、図を描くように直感的に組み立てることができる。これにより、複雑なプログラムを書かなくても、一連の作業手順を整理することが可能になる。

また、Elyra は Kubernetes や Kubeflow Pipelines といった分散処理基盤とも連携が可能である。Kubernetes はコンテナ化されたアプリケーションの配置や増減を自動的に管理するためのツールで、Kubeflow Pipelines は機械学習ワークフローを Kubernetes 上で自動化・管理するためのフレームワークである。これらを組み合わせることで、ローカル PC で構築、テストした機械学習ワークフローを、クラウド環境でも同じ設定で実行できる。さらに、Kubernetes の機能により、CPU 使用率やメモリ使用率に基づいて自動的にコンテナ数を増やし、処理完了後は自動的にリソースを解放するため、コスト効率と処理速度の両方を最適化できる。

本報告では、まず Quakeflow で提案されている地震波形データ処理のワークフローを、Elyra を使って「データのダウンロード」「前処理」「Phasenet の実行」などの工程ごとに Jupyter Notebook ファイルとして分割・部品化し、それらを Elyra 上で組み合わせて一連の流れとして再構築した。

さらに、日本の地震波形データを処理できるようにするために、データのダウンロードや前処理の工程を日本版にカスタマイズした Docker イメージや Jupyter Notebook を作成し、処理ワークフローの動作の検証を行った。

詳細については当日の発表の際に発表用 PC とクラウド上に Elyra 実行環境を構築し、デモンストレーションを用いて発表する。

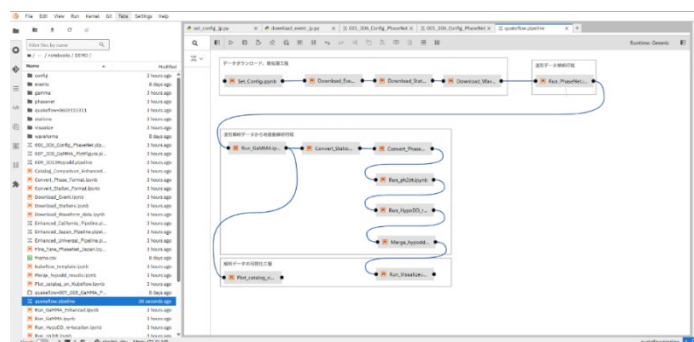


図 1 Elyra Pipeline Editor 画面 - 各ステップがビジュアルに配置されたパイプライン構成

参考文献

1. Zhu, W. & Beroza, G. C., "PhaseNet: a deep-neural-network-based seismic arrival-time picking method," Geophys. J. Int., 216(1), 261–273, 2019. <https://academic.oup.com/gji/article/216/1/261/5129142>
2. Zhu, W., McBrearty, I. W., Mousavi, S. M., Ellsworth, W. L., & Beroza, G. C., "Earthquake Phase Association Using a Bayesian Gaussian Mixture Model," J. Geophys. Res. Solid Earth, 127, e2021JB023249, 2022. <https://doi.org/10.1029/2021JB023249>
3. Zhu, W. et al., "QuakeFlow: a scalable machine-learning-based earthquake monitoring workflow with cloud computing," Geophys. J. Int., 232(1), 684–693, 2023. <https://doi.org/10.1093/gji/ggac216>
4. 防災科学技術研究所, NIED Hi-net, 2019. <https://doi.org/10.17598/NIED.0003> (<https://doi.org/10.17598/NIED.0003>)