

電波空間情報のための探索的データ解析クラウドプラットフォームの インタフェース実装についての一検討

鈴木 博貴[†] 脇坂 優樹[†] 川喜田佑介[†] 市川 晴久[†]

[†] 電気通信大学

An Examination on Implementation of Interface of Exploratory Data Analysis Cloud Platform for Radio Space Information

Hiroataka SUZUKI[†], Yuuki WAKISAKA[†], Yuusuke KAWAKITA[†], and Haruhisa ICHIKAWA[†]

[†] The University of Electro-Communications

1 はじめに

電波の時間的・空間的に空いている周波数帯であるホワイトスペースを利用することで周波数資源の枯渇への対策となる。そのホワイトスペースの利用状況調査のために、複数拠点にて電波空間そのもののサンプリングを行い、収集し、蓄積した情報をもとにデータベース (DB) の構築が行われている [1]。また、無線規格に非依存な共通ハードウェアを用いて複数の無線規格の電波が含まれる広範囲の周波数帯域をサンプリングし、規格に適したソフトウェア受信機を立ち上げて処理を行うことで、新たな規格や独自開発の規格の導入に際してソフトウェア受信機の追加のみで対応を可能とする Appliance Defined Ubiquitous Network: ADUN [2] がある。川喜田らは、ADUN における課題として電波空間情報に含まれる信号の無線規格を検出することによって、ソフトウェア受信機を自動的に起動する機能の実現を挙げている [3]。いずれの場合も電波空間情報を解析し特定の情報を得るために閾値の設定や特徴量の抽出、選択が必要である。また、このような解析処理は試行的に繰り返される。加えて、近年注目を集める Internet of Things (IoT) のサービスの一つとしてセンサーによる実世界情報の収集・解析を行い、その結果がダッシュボードによって提供されている。このようなインタフェース (I/F) は利用ユーザに対して複数の情報を一括して視覚的に得ることが可能であり、データに関する洞察を与えるために有効である。

解析処理においては、目的の結果を得るための解析手法の選択や解析時のパラメータの決定、対象データの絞り込みのために、データに対する洞察が得られていることが望ましい。特に解析対象とするデータに対する洞察を、主に視覚的に得るための手法として探索的データ解析 (Exploratory Data Analysis: EDA) がある。EDA は解析初期の段階においてデータを俯瞰的にさまざまな角度から観察してその洞察を得るために有効な手法であるため、一つのデータに対して繰り返し解析処理を行えることが望ましい。したがってその対象を電波空間情報をする場合は、過去に遡ってのデータの利用や、同一データに対する複数回の解析、別地点で収集した電波空間の情報を組み合わせての解析への利用を想定して蓄積・検索が行える必要がある。

電波空間情報の蓄積や解析には、電波空間情報の高サンプリングレートに起因するデータサイズの大きさが問題となる。蓄

積においては大容量かつ高速な I/O を実現するストレージが、解析においては高い処理能力が必要であることから、蓄積・解析の機能をクラウド上に配置することが適切である。

クラウド上に配置した蓄積と解析の機能を利用して EDA を行う際にはユーザへの解析結果の視覚的な提供のほか、解析処理内容やパラメータ変更を柔軟かつインタラクティブに行うためのユーザ I/F を必要とする。筆者らは電波空間情報への EDA が適用可能なシステムの実現を目的として、解析の処理内容やパラメータ変更操作におけるインタラクティブ性とそれを損なわない処理性能、解析対象とする電波空間情報へのアクセシビリティを向上させる検索機能。加えて計算機資源の効率的利用のための負荷分散を要件としたクラウドプラットフォームを構想した。本稿では、プラットフォームの構想案とクライアント・クラウド間の I/F の実装について検討した結果を報告する。

2 電波空間情報のための探索的データ解析クラウドプラットフォームの構想

2.1 プラットフォームの構想案

大きくは 3 つの要素からなる。(A) 処理対象データ：蓄積された電波空間情報ファイルとそのメタデータ DB。(B) プロセッサ：ロードバランサと信号処理および電波空間情報検索プロセッサ、それらの機能をクライアントに公開するプロセッサ I/F。(C) ユーザ I/F：利用者に提供される解析処理の記述と実行の要求、パラメータ設定、処理結果の表示機能。(A)、(B) がクラウド上に存在し、(C) からの要求に応じて (B) を通してメタデータを利用したファイルの検索と読み出す。読み出したデータに対する解析処理を実行し、その結果をリアルタイムに (C) へと送信し、要求に応じた形式で表示する。本プラットフォームでは信号処理が主たる機能となるため、この機能のベースとして GNU Radio Companion [4] を選定し、用いることとした。選定理由については、以前文献 [5] にて報告している。

2.2 GNU Radio Companion

GRC は GNU Radio (GR) によるフローグラフと呼ばれる信号処理フローのプログラムを GUI を用いて構築可能なエディタである。図 1 に GRC の構造を示す。GR にて提供されているコンポーネントは「ブロック」という単位で扱われる。ブロックは C++ で記述されており、SWIG を通して Python か

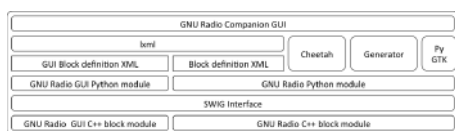


図1 GNU Radio Companion の基本構造

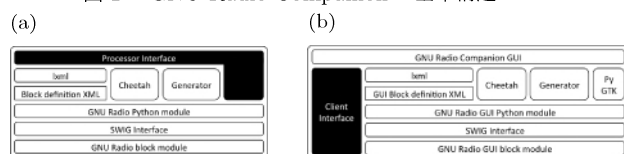


図2 機能分割による (a) 信号処理系、(b) GUI 系の構造

ら利用される。このブロックにはデータの出入力ポートが定義されており、それらのポートをつなぎ合わせ、ブロックを複数接続することでフローグラフを構築する。GRC では各ブロックについての情報が定義された XML ファイルの情報をもとにグラフィカルなブロックを表示している。この XML ファイルとフローグラフのテンプレートを元に、Generator においてテンプレートエンジンである Cheetah を用いてフローグラフの Python コードを生成し、それを実行する。

2.3 GRC をプラットフォームのベースとする際の課題

GRC をプラットフォームのベースとして適用するためには、GRC のユーザ I/F と信号処理系を切り離すことで GRC のフローグラフ構築とコード生成機能をリモートから利用可能とすること。また、蓄積した電波空間情報ファイルのメタデータを用いての検索機能の実現が課題とである。GRC を信号処理系の機能を含む 2.1 の (B) プロセッサ側 (図 2(a)) と、ユーザ I/F と GR の GUI 機能を含む 2.1 の (C) ユーザ I/F 側 (図 2(b)) に機能を分割し、分離した機能の渡しとなる I/F (図 2 の黒色部) を追加実装することでリモートからの機能利用を満たすこと。また、GRC にはファイルのメタデータについてユーザが既知である前提に基づくファイル選択機能しか有さないため、新たに蓄積される電波空間情報のファイルのメタデータによるメタデータ DB の構築と、検索用の I/F によってメタデータに基づいたファイル検索機能の提供、検索結果のファイルを選択して信号処理系に読み込ませる機能を実装することによりメタデータによる検索機能の実現を満たすことによって課題を解決を図る。以降では、「プロセッサ」と「ユーザ I/F」を接続する I/F 実装の課題を中心として次に論ずる。

3 プロセッサ-ユーザ I/F 間の I/F の実装

以降では、図 2 の (a) を「プロセッサ GRC」、(b) を「クライアント GRC」とし、プロセッサ GRC で生成される信号処理と解析を行うフローグラフの Python コードを「プロセッサフローグラフ」、クライアント GRC で生成される解析処理結果を表示、解析パラメータ変更操作を受け付ける Python コードを「クライアントフローグラフ」とする。2.3 で述べた通り、図 2 のように GRC を機能分割する。機能を分離した上で主に必要となる機能は、プロセッサ-ユーザ I/F 間の通信機能とその制御、フローグラフファイルからのコード生成時にプロセッサ側とユーザ I/F 側のそれぞれで必要な機能を有するコードを生成するようにすること、の二点である。

3.1 プロセッサ-ユーザ I/F 間の通信機能とその制御

プロセッサ GRC とクライアント GRC でのフローグラフの編集・実行・停止などのコマンド命令、クライアントフローグラフとプロセッサフローグラフ間のパラメータ変更などは確実な到達性を考慮して TCP を、プロセッサフローグラフからクライアントフローグラフへの解析処理結果の送信にはリアルタイム性を重視して UDP をトランスポートプロトコルに用いることを検討している。

クライアント GRC でのフローグラフ編集やコマンド命令お

よび、クライアントフローグラフの GUI 上でのパラメータ操作は、その操作のハンドラのラッパーを実装してプロセッサ側に変更を通知する方針である。

今回はユーザ I/F のクライアントプログラムをプラットフォームのベースとした GRC に限定しているが、HTML5、CSS3、JavaScript を用いて必要な機能を再実装すれば、ブラウザ上での利用も可能である。その場合、UDP をトランスポートプロトコルとして使用できないため、クライアント GRC では全ての通信に WebSocket を利用することも検討される。

3.2 フローグラフファイルから生成されるコードの制御

本来の GRC では、信号処理系と GUI 系は同一計算機上で動作するため、一つのフローグラフファイルに両方の機能が記述され、両方の機能を有するフローグラフを生成する。一方、今回の実装においては、一つのフローグラフファイルから生成されるコードはプロセッサ側では信号処理、ユーザ I/F 側では処理結果表示の GUI、と異なること、またそれぞれの側で実行される Python コード間の通信機能を含める必要があることから、それぞれの Generator におけるコード生成時にフローグラフファイル中の必要な機能のみを含むフローグラフのコードを生成する必要がある。それに加えて、フローグラフ上の変数と処理結果の出力先の GUI の対応をクライアントフローグラフとプロセッサフローグラフで取らなければならない。

それぞれで必要な機能を含むフローグラフのコード生成には Cheetah で読み込むテンプレートの書き換えで対処可能である。変数や出力先の管理、対応付けについてはそれぞれの Generator での生成されたコードの静的解析結果の整合性を検証する機能を実装することによる解決を検討している。

4 ま と め

本稿では、解析処理を効率的にする探索的データ解析の電波空間情報への適用を可能とするシステムの実現を目的としてクラウドベースの電波空間情報解析プラットフォームを構想し、その構成案を示すとともに主たる機能となる信号処理のベース実装として GRC を用いる際の課題について整理した。加えて、GRC をベースとする際の課題の一つについて、GRC の機能分割による解決と、その際に必要となる機能要件と実装についての検討を行った。

プロセッサ-ユーザ I/F 間の I/F の実装に関して、通信についてはトランスポートプロトコルの検討のみに留まったため、それより上位のプロトコルについて必要な機能の検討が不十分である。また、利用ユーザごとのユーザ空間の管理方法についても検討の必要がある。以上のことに今後取り組み、プロセッサ-ユーザ I/F 間の I/F の実装の実現をまず目指していく。

謝 辞

本研究は JSPS 科研費 25240010, 25730054 の助成を受けた。

文 献

- [1] T. Taher, et al. Global spectrum observatory network setup and initial findings. In *Proc. CROWNCOM 2014*, pp. 79–88, 2014.
- [2] H. Ichikawa, et al. Ubiquitous networks with radio space extension over broadband networks. *IEICE Trans. on Comm.*, Vol. E90-B, No. 12, pp. 3445–3451, Dec 2007.
- [3] 川喜田ほか. 自立給電型ワイヤレスセンサのためのユビキタス・センシング・プラットフォーム. 計測と制御, Vol. 52, No. 11, pp. 966–972, Nov 2013.
- [4] Gnu radio companion - gnu radio - gnuradio.org. <https://gnuradio.org/redmine/projects/gnuradio/wiki/GNURadioCompanion>.
- [5] 鈴木ほか. B-18-21 電波空間情報のための探索的データ解析 cloud platform の課題. 電子情報通信学会ソサイエティ大会講演論文集, Vol. 2015, No. 2, p. 355, Sep 2015.