サーバ上におけるベクトルタイルの画像タイル化に関する検討

実施期間　令和 3 年度
地理空間情報部情報普及課　川井　拓弥　淺田　剛史
大矢　和生　渡部　金一郎

1. はじめに

国土地理院では，現在，画像形式で提供しているウェブ地図用のタイルデータをベクトル形式で提供すべく，ベクトルタイルの提供実験を行っている．

今後，ベクトルタイル方式によるウェブ地図データの提供が主流となった場合，画像タイルのみに対応しているウェブ地図や GIS においてこれを利用するためには，ベクトルタイルを画像タイル化する必要が生じる可能性がある．そこで，このようなウェブ地図や GIS の開発者又は利用者が必要に応じて画像タイルを作成できるよう，任意のサーバ上においてベクトルタイルを画像タイル化する仕組みの検討及び実証を行った．

2. サーバ上における画像タイル化機能の開発

ベクトルタイルを描画できるオープンソースのウェブ地図用 JavaScript ライブラリ「Mapbox GL JS」と，コマンドライン上でウェブブラウザを動作させることができるオープンソースの Node.js ライブラリ「Playwright」を用いて，Linux 系 OS で動作しているサーバ上に画像タイルレンダリング機能を実装するとともに，一度作成した画像タイルを再利用することで描画処理の負荷を減らし，地図表示を高速化するため，画像タイルキャッシュ機能を実装した（図-1）．

2.1 画像タイルレンダリング機能

画像タイルレンダリング機能は，Playwright によって動作するウェブブラウザ上で，Mapbox GL JS Style Specification に則ったスタイルファイルに基づいて Mapbox GL JS によりベクトルタイルを描画し，それを画像タイル化する機能である．

ベクトルタイルの描画処理は，通常，クライアント側のウェブブラウザで行われるが，本機能の場合，ベクトルタイルをサーバ上にある Playwright の仮想ブラウザで描画した上で，画面キャプチャを行っている．すなわち，あるタイル 1 枚分の描画を行い，それをキャプチャして PNG 形式のファイルを作成する処理を繰り返す仕組みとなっている．

画像タイルを表示する既存のウェブ地図での読込を容易にするため，作成される画像タイルのタイル座標の付与方式及びファイル名の命名方式は，地理院タイルに準じた．また，画像タイルのファイル形式は PNG 形式とし，拡張子は「.png」とした．

画像タイルは，以下のような URL により行われるクライアントからのタイルデータ取得のリクエストに応じて作成されるようにした．

http://*servername*/xyz/*stylename*/{z}/{x}/{y}.png

（斜字にした部分には，実際にはサーバのドメイン名，スタイルファイル名が入る．）

2.2 画像タイルキャッシュ機能

画像タイルキャッシュ機能は，2.1.に示した機能によって作成した画像タイルをサーバ内にキャッ

シュとして保存するとともに，クライアントからのリクエストを受けた際，サーバ内に画像タイルのキャッシュが存在する場合はそれをクライアントに返し，キャッシュが存在しない場合は 2.1 に示した機能によって新たに作成した画像タイルをクライアントに返す機能である．

また，管理者向けの機能として，保存されたキャッシュをファイル単位又はフォルダ単位で削除する機能も併せて実装した．

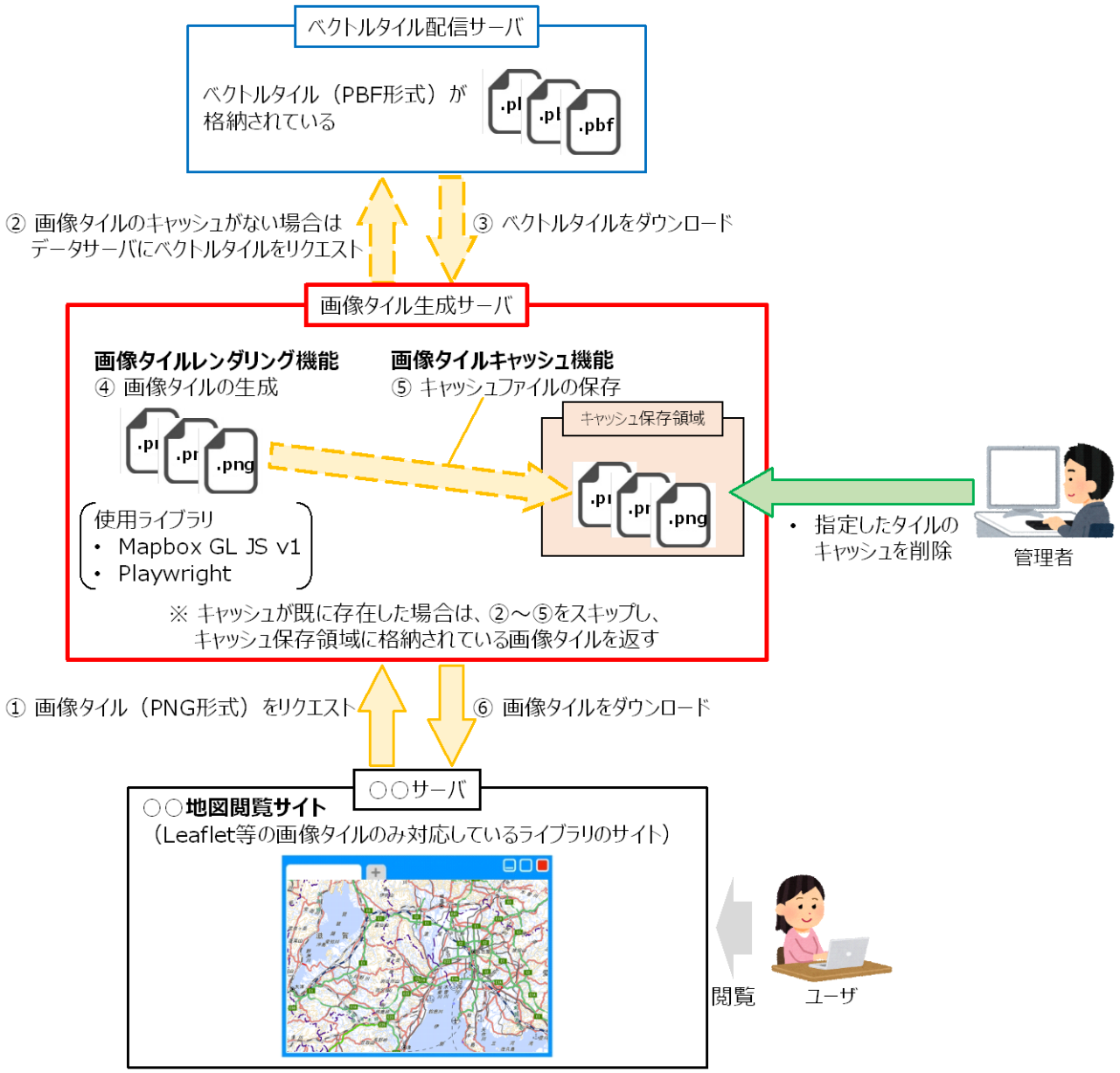
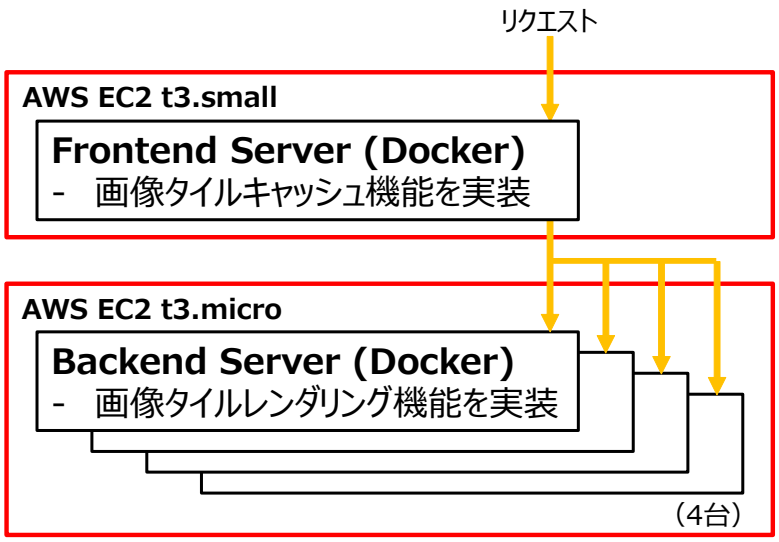


図-1　画像タイル化機能の概要

3. 動作実証

画像タイル化機能を実装したサーバにリクエストを送信し，当該サーバで作成され配信される画像タイルを表示するテストサイトを構築し，地図の描画及び表示速度について検証した．なお，当該サーバは，画像タイルレンダリング機能を実装した仮想サーバ 4 台と，画像タイルキャッシュ機能を実装した仮想サーバ 1 台で構成した（図-2）．



※ 仮想サーバにはいずれも nginx/1.18.0 を利用

図-2　動作実証に用いたサーバの構成

その結果，Mapbox GL JS によりベクトルタイルをクライアント側のウェブブラウザで描画した場合と同様の地図が描画され，画像タイルの内容そのものに特段の問題はなかったが，各タイルの表示完了まで 5～30 秒ほどかかり，ウェブ地図の閲覧性としてはベクトルタイルをクライアント側のウェブブラウザで描画した場合に比べてかなり劣る結果となった（図-3）.

なお，Playwright の仮想ブラウザとして Chromium 及び Firefox を設定して検証したが，描画性能に特段の差は見受けられなかった.

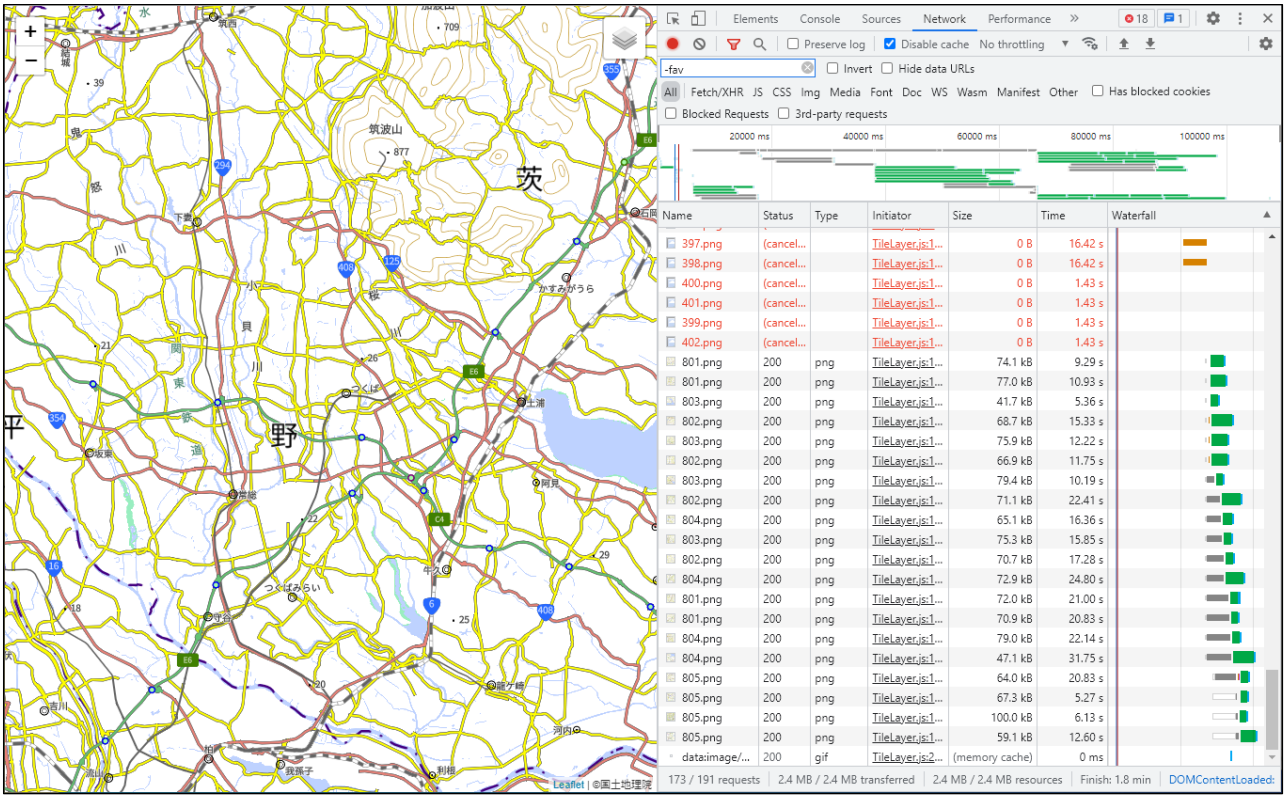


図-3　画像タイルの表示に要する時間の確認結果の一例

4. 今後の課題

今回の検討の結果，リクエストに応じてサーバ上でベクトルタイルを画像タイル化し，これを配信することは技術的に可能であるものの，ベクトルタイルをクライアント側のウェブブラウザで描画する場合に比べて，地図の表示にかなり時間がかかることを確認した．別途検討中であるベクトルタイルの最適化・軽量化の取組を進めることにより，今回検討した仕組みにおける描画処理の速度を改善することは不可能ではないが，その効果は限定的であると考えられる．やはり，ベクトルタイルをクライアント側のウェブブラウザで直接描画した方がユーザフレンドリーであると考えられることから，これを促進する技術的な支援を行っていくことが妥当であると考えられる．

参考文献

Mapbox: Mapbox GL JS, https://docs.mapbox.com/mapbox-gl-js/guides/ (accessed 23 Mar. 2022).

Mapbox: Style Specification, https://docs.mapbox.com/mapbox-gl-js/style-spec/ (accessed 23 Mar. 2022).

Microsoft: Playwright, https://playwright.dev/ (accessed 23 Mar. 2022)

The United Nations Vector Tile Toolkit: plow-http (Initial test of raster-rendering server), https://github.com/unvt/plow-http (accessed 23 Mar. 2022)