

# Nghiên cứu và xây dựng phần mềm máy chủ dịch vụ bản đồ theo tiêu chuẩn Vector Tile

Phạm Mạnh Đạt<sup>1</sup>, Trần Trung Chuyên<sup>2</sup>  
<sup>1,2</sup> Trường Đại học Mở - Địa Chất

---

## TÓM TẮT

Mục đích và phạm vi nghiên cứu của bài báo là xây dựng một máy chủ hỗ trợ dịch vụ bản đồ theo tiêu chuẩn MapBox Vector Tile với các nền tảng mã nguồn mở, tối ưu hoá hiệu năng cao, dễ dàng triển khai tích hợp cho các ứng dụng nền tảng phổ biến như WebGIS, MobileGIS và DesktopGIS. Đối tượng và phương pháp nghiên cứu sử dụng nền tảng NodeJS, hệ quản trị cơ sở dữ liệu không gian PostGIS/PostgreSQL và dựa trên những tiêu chuẩn kỹ thuật mở Vector Tile của MapBox đưa ra.

Kết quả chính của bài báo là xây dựng một giải pháp với chi phí thấp, tận dụng được các công nghệ mã nguồn mở có sẵn, cho việc xây dựng một máy chủ dịch vụ bản đồ số theo tiêu chuẩn Vector Tile. Phù hợp với những dự án vừa và nhỏ, kinh phí hạn chế và thời gian triển khai nhanh chóng, hiệu suất xử lý dữ liệu cao với tài nguyên phần cứng không yêu cầu quá cao. Và đặc biệt là dễ dàng chia sẻ dịch vụ với các nền tảng khác nhau Web, Mobile và Desktop.

*Từ khóa: Vector Tile; Tile Map Services; GIS; OpenSource;*

---

## 1. Đặt vấn đề

Hiện nay, các dịch vụ truyền tải và chia sẻ dữ liệu bản đồ số đang được sử dụng phổ biến như là Web Map Service (WMS), Tile Map Service (TMS)... Với cách thức truyền tải dữ liệu thông qua một máy chủ dịch vụ bản đồ như MapServer, GeoServer, ArcGIS Server... Máy chủ (Server) sẽ nhận các yêu cầu của máy khách (Client) và kết xuất (Render) hình ảnh (Raster Tile) sau đó trả kết quả về cho máy khách hiển thị trên giao diện người dùng. Nhược điểm của các dịch vụ này là dữ liệu hình ảnh trả về sẽ không thể có đầy đủ các thông tin thuộc tính và không gian của các đối tượng địa lý, máy chủ cần cấu hình khá cao để có thể kết xuất hình ảnh ổn định, việc thay đổi kiểu dáng (Style) hiển thị sẽ phụ thuộc hoàn toàn vào phía máy chủ nên đối với loại dữ liệu lớn, số lượng máy khách truy cập nhiều sẽ cần nguồn tài nguyên cấu hình máy chủ rất cao.

Để khắc phục những nhược điểm trên, hãng công nghệ bản đồ số MapBox đã đưa ra một tiêu chuẩn Vector Tile, đây là một tiêu chuẩn dữ liệu hoàn toàn mở được công bố trên kho lưu trữ GitHub. Cho tới hiện tại, tiêu chuẩn MapBox Vector Tile đã được cộng đồng bản đồ số công nhận như là một tiêu chuẩn chung, mã nguồn mở cho dịch vụ bản đồ định dạng Vector Tile.

Bài báo sẽ giúp người đọc hiểu hơn về tiêu chuẩn dịch vụ bản đồ Vector Tile và tính ứng dụng cùng những ưu điểm vượt trội hơn so với các dịch vụ bản đồ Raster Tile. Dựa trên những tiêu chuẩn kỹ thuật mở của MapBox đưa ra, cùng các nền tảng mã nguồn mở NodeJS, PostGIS/PostgreSQL và với nội dung được trình bày trong bài báo sẽ giúp người đọc có thể tự xây dựng một dịch vụ bản đồ tùy chỉnh theo chuẩn Vector Tile cho riêng mình.

## 2. Cơ sở lý thuyết và nội dung nghiên cứu

### 2.1. Cơ sở lý thuyết

Dựa vào tiêu chuẩn mở MapBox Vector Tile (MVT), bài báo sẽ tập trung nghiên cứu để có thể xây dựng một phần mềm dịch vụ bản đồ theo chuẩn Vector Tile chạy trên nền tảng NodeJS qua giao thức HTTP cùng các tiêu chuẩn dịch vụ Rest API và máy chủ cơ sở dữ liệu không gian sử dụng là PostgreSQL/PostGIS.

Vector tile được mã hóa dưới định dạng Google Protobufs (PBF), cho phép tuần tự hóa dữ liệu có cấu trúc. Mapbox Vector Tile sử dụng .MVT làm phần mở rộng của tệp tin. Các chi tiết đặc tả phần lớn được cấu trúc xung quanh các quy tắc được triển khai dựa trên cơ sở tệp tin Google Protobufs.

#### 2.1.1 Cấu trúc mã hóa hình học của MapBox Vector Tile

Để mã hóa thông tin địa lý thành các vector tile, cần phải chuyển đổi tọa độ địa lý vĩ độ và kinh độ thành tọa độ lưới ô vuông theo cấu trúc chia tile. Các ô vector tile không có khái niệm về thông tin địa lý. Sau đó, mã hóa các điểm, đường và vùng thành các cặp x, y theo các mức thu phóng của bản đồ, thông thường là từ 0 đến 22. Một vector tile có thể lưu nhiều lớp (layer), mỗi lớp lại có thể chứa nhiều đối tượng điểm, đường và vùng khác nhau.

### 2.1.2 Cấu trúc mã hóa thuộc tính của MapBox Vector Tile

Các thuộc tính được mã hóa thành một chuỗi theo Tags tồn tại trong một đối tượng vector có các giá trị số nguyên tham chiếu Keys và Values chỉ định các cặp key:value ban đầu từ hình học. Đối với lớp dữ liệu lớn, điều này sẽ loại bỏ sự dư thừa đối với các thuộc tính có cùng khóa và giá trị tương tự, nhằm giảm dung lượng lưu trữ tile

#### Geojson

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "geometry": { ... },
      "type": "Feature",
      "properties": {
        "hello": "world",
        "h": "world",
        "count": 1.23
      }
    },
    {
      "geometry": { ... },
      "type": "Feature",
      "properties": {
        "hello": "again",
        "count": 2
      }
    }
  ]
}
```

#### Vector tile

```
layers {
  version: 2
  name: "points"
  features: {
    id: 1
    tags: 0
    tags: 0
    tags: 1
    tags: 0
    tags: 2
    tags: 1
    type: Point
    geometry: ...
  }
  features {
    id: 2
    tags: 0
    tags: 2
    tags: 2
    tags: 3
    type: Point
    geometry: ...
  }
  keys: "hello"
  keys: "h"
  keys: "count"
  values: { string_value: "world" }
  values: { double_value: 1.23 }
  values: { string_value: "again" }
  values: { int_value: 2 }          extent: 4096
}
```

## 2.2. Nội dung nghiên cứu

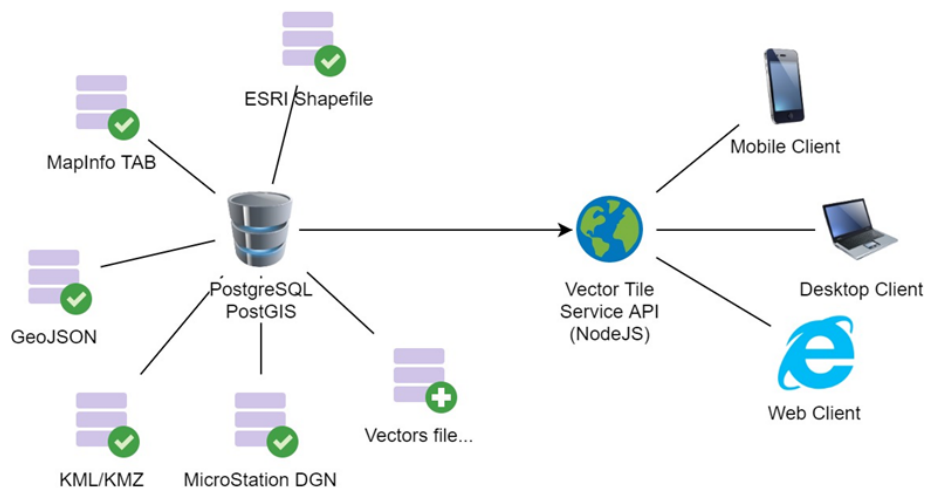
### 2.2.1. Cài đặt môi trường phát triển thử nghiệm

Hệ thống có thể xây dựng và chạy trên đa nền tảng như Windows, Linux và MacOS. Trong phạm vi bài báo sẽ sử dụng môi trường phát triển thử nghiệm như sau:

- Hệ điều hành: Ubuntu 18.04.2
- Phiên bản PostgreSQL: 9.5
- Phiên bản PostGIS: 2.5
- Phiên bản NodeJS: 12.16.1

### 2.2.2. Kiến trúc xây dựng hệ thống

Hệ thống sử dụng nền tảng NodeJS tạo máy chủ dịch vụ web kết nối tới cơ sở dữ liệu PostgreSQL, dựa vào phần mở rộng PostGIS để xử lý các dữ liệu không gian, tạo dữ liệu định dạng theo chuẩn MapBox Vector Tile và chia sẻ qua Rest API cho các nền tảng máy khách.



### 2.2.3. Xây dựng máy chủ dịch vụ bản đồ Vector Tile

Giả thiết môi trường thử nghiệm đã được cài đặt NodeJS và PostgreSQL/PostGIS. Người đọc có thể tham khảo cách cài đặt tại đây:

- ✓ Cài đặt NodeJS:

<https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-ubuntu-18-04>

- ✓ Cài đặt PostgreSQL/PostGIS:

<https://computingforgeeks.com/how-to-install-postgis-on-ubuntu-debian/>

Thiết lập kết nối từ NodeJS tới PostgreSQL bằng cách sử dụng thư viện 'pg' trên kho lưu trữ NPM.

```
const { Client } = require('pg');
var connectionString =
  "postgres://postgres:postgres@localhost:5432/database";

const client = new Client({
  connectionString: connectionString
});
```

Tạo một file package.json để lưu các cấu hình gói thư viện cần sử dụng là 'pg' và 'express'.

```
const express = require('express');
const { Client } = require('pg');
const connectionString =
  'postgres://postgres:postgres@localhost:5432/database';

const client = new Client({
  connectionString: connectionString
});

client.connect();

var app = express();

app.set('port', process.env.PORT || 3000);
```

```

{
  "name": "vector_tile_service",
  "version": "1.0.0",
  "description": "Demo vector tile services",
  "main": "app.js",
  "dependencies": {
    "express": "^4.16.4",
    "pg": "^7.8.2"
  }
}

```

Tạo một file **index.js** để kiểm tra kết nối và truy vấn câu lệnh SQL tới PostgreSQL/PostGIS.

```

app.get('/', function (req, res, next) {
  client.query('SELECT * FROM test_table where id = $1', [1],
  function (err, result) {
    if (err) {
      console.log(err);
      res.status(400).send(err);
    }
    res.status(200).send(result.rows);
  });
});

app.listen(3000, function () {
  console.log('Server is running.. on Port 3000');
});

```

Viết hàm thủ tục tạo câu lệnh truy vấn SQL với các tham số đầu vào là X,Y,Z theo chuẩn Tile Map Service (TMS)

```

const sql = (params, query) => {
  let bounds = merc.bbox(params.x, params.y, params.z, false, '900913')

  return `
SELECT
  ST_AsMVT(q, '${params.table}', 4096, 'geom')

FROM (
  SELECT
    ${query.columns ? `${query.columns},` : ''}
    ST_AsMVTGeom(
      ST_Transform(${query.geom_column}, 3857),
      ST_MakeBox2D(ST_Point(${bounds[0]}, ${bounds[1]}), ST_Point(${
    bounds[2]
  }, ${bounds[3]}))
    ) geom

  FROM (
    SELECT
      ${query.columns ? `${query.columns},` : ''}
      ${query.geom_column},
      srid
    FROM
      ${params.table},
      (SELECT ST_SRID(${query.geom_column}) AS srid FROM ${
    params.table
  } LIMIT 1) a

    WHERE
      ST_transform(
        ST_MakeEnvelope(${bounds.join()}, 3857),
        srid
      ) &&
      ${query.geom_column}

    -- Optional Filter
    ${query.filter ? `AND ${query.filter}` : ''}
  ) r
) q
`
}

```

Để tạo một dịch vụ theo chuẩn Tile Map Service cần cấu hình thêm một route với cấu trúc đường dẫn truy cập như sau: '/mvt/:table/:z/:x/:y', Trong đó:

- /mvt/* là đường dẫn route gốc của máy chủ http cho dịch vụ
- /:table/* là tên bảng dữ liệu không gian, hỗ trợ cả Point, Polygon và Line
- /:z/* là mức zoom, kiểu số nguyên từ 1 tới n
- /:x/* là cột x, kiểu số nguyên
- /:y/* là hàng y, kiểu số nguyên

Tạo một module cho dịch vụ TMS như sau:

```
module.exports = function(app, opts, next) {
  app.route({
    method: 'GET',
    url: '/mvt/:table/:z/:x/:y',
    schema: schema,
    handler: function(request, reply) {
      app.pg.connect(onConnect);

      function onConnect(err, client, release) {
        if (err)
          return reply.send({
            statusCode: 500,
            error: 'Internal Server Error',
            message: 'unable to connect to database server'
          });

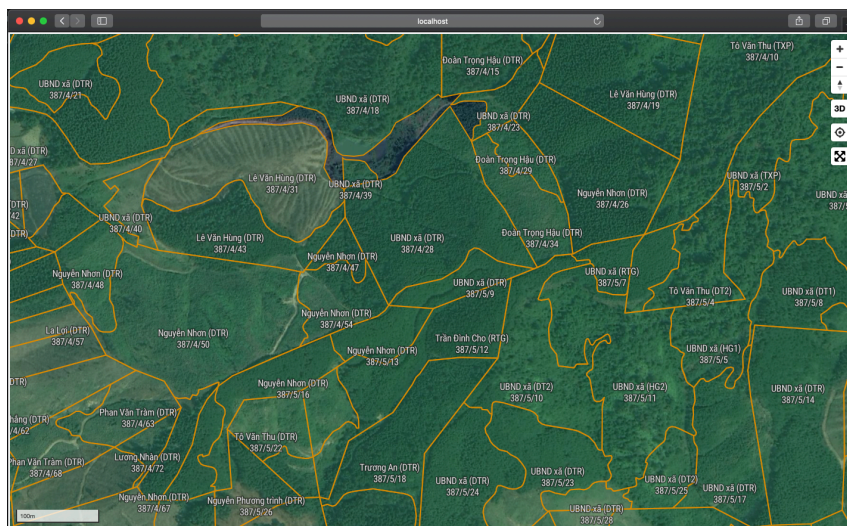
        client.query(sql(request.params, request.query), function onResult(
          err,
          result
        ) {
          release();
          if (err) {
            reply.send(err)
          } else {
            const mvt = result.rows[0].st_asmvt;
            if (mvt.length === 0) {
              reply.code(204)
            }
            reply.header('Content-Type', 'application/x-protobuf').send(mvt)
          }
        })
      }
    }
  });
  next();
};
```

Sử dụng lệnh ‘`node ./index.js`’ để chạy thử nghiệm mã, dịch vụ bản đồ sẽ chạy theo giao thức HTTP ở địa chỉ: `http://localhost:3000/mvt/{table}/{z}/{x}/{y}`

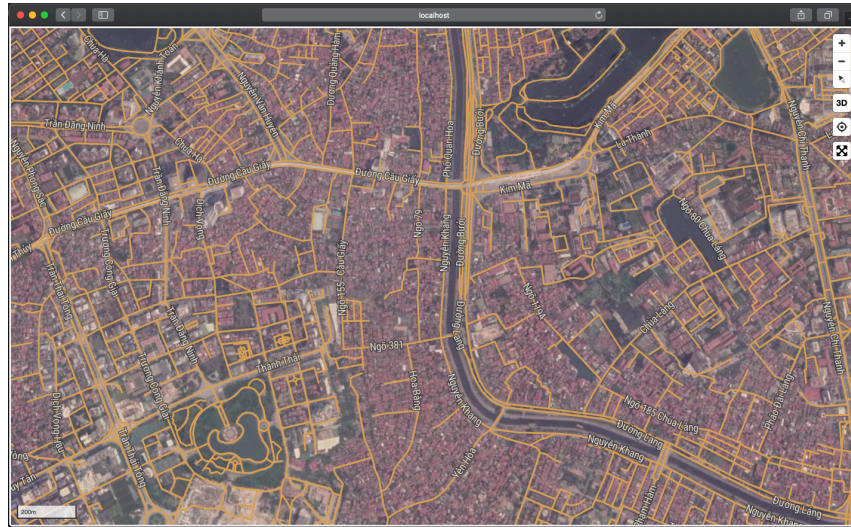
GET /v1/mvt/{table}/{z}/{x}/{y} trả về MVT	
Trả về dữ liệu Mapbox Vector Tile (MVT). Tên của layer là tên của bảng dữ liệu không gian.	
Tham số	
Tên tham số	Mô tả
<code>geom_column</code> <i>string</i> ( <i>query</i> )	Cột geometry của bảng <i>Mặc định</i> : geom
<code>columns</code> <i>string</i> ( <i>query</i> )	Tùy chọn các cột dữ liệu sẽ trả về, Mặc định trống
<code>filter</code> <i>string</i> ( <i>query</i> )	Tùy chọn lọc theo câu lệnh SQL mệnh đề WHERE
<code>table</code> * <i>required</i> <i>string</i> ( <i>path</i> )	Tên của bảng dữ liệu (bắt buộc)
<code>z</code> * <i>required</i> <i>integer</i> ( <i>path</i> )	Z giá trị của ZXY tile.
<code>x</code> * <i>required</i> <i>integer</i> ( <i>path</i> )	X giá trị của ZXY tile.
<code>y</code> * <i>required</i> <i>integer</i> ( <i>path</i> )	Y giá trị của ZXY tile.

### 3. Kết quả và ứng dụng

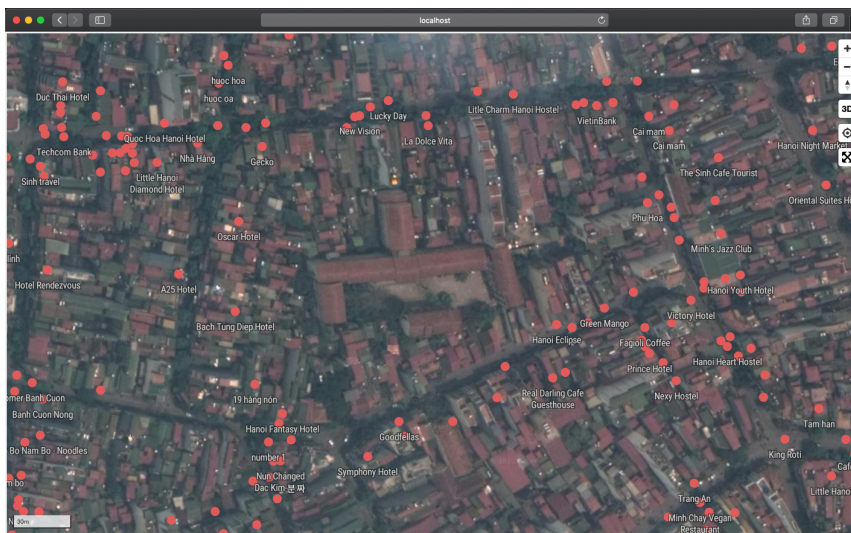
Để tích hợp và sử dụng dịch vụ Vector Tile, có thể sử dụng thư viện MapBox GL. Đây là một thư viện nguồn mở hỗ trợ đa nền tảng WebGIS, MobileGIS và DesktopGIS, được cung cấp miễn phí trên kho GitHub của MapBox, hỗ trợ đầy đủ các tiêu chuẩn đặc tả của tiêu chuẩn định dạng Vector Tile.



Hình 1. Hiện thị vector tile dạng vùng trên WebGIS



Hình 2. Hiện thị vector tile dạng đường trên WebGIS



Hình 3. Hiện thị vector tile dạng điểm trên WebGIS

## 4. Kết luận

### 4.1 Đánh giá kết quả đạt được

Bài báo đã hoàn thành được mục tiêu đề ra, dựa trên những nền tảng công nghệ thông tin được được nêu lên, bài báo đã bước đầu xây dựng được giải pháp giúp xây dựng nhanh chóng phần mềm máy chủ bản đồ theo tiêu chuẩn vector tile, chi phí rẻ với việc hoàn toàn sử dụng các nền tảng công nghệ mã nguồn mở, người dùng có thể chủ động tùy biến dịch vụ bản đồ cho từng dự án cụ thể, dễ dàng triển khai tích hợp cho các ứng dụng nền tảng phổ biến như WebGIS, MobileGIS và DesktopGIS.

### 4.2 Hướng phát triển tiếp theo của bài báo

Trong nội dung của bài báo xây dựng máy chủ dịch vụ bản đồ được phát triển dựa trên các nền tảng mã nguồn mở, phần chính của dịch vụ được xây dựng với NodeJS, tuy nhiên hoàn toàn có thể mở rộng phát triển cho những ngôn ngữ lập trình khác như PHP, Java, Python... để tăng tính linh hoạt, dễ tích hợp để sử dụng được cho đa dạng hơn các loại dự án trong phát triển hệ thống chia sẻ và khai thác dữ liệu bản đồ.

**Tài liệu tham khảo**

1. MapServer introduction. <http://mapserver.org>
2. GeoServer Documentation. <https://docs.geoserver.org>
3. PostgreSQL Documentation. <http://www.postgresql.org>
4. NodeJS Documentation. <https://nodejs.org/en/docs/>
5. MapBox Vector Tile specification. <https://docs.mapbox.com/vector-tiles/specification/>



# Develop a map server web service based on vector tile

Pham Manh Dam, Tran Trung Chuyen

## SUMMARY

The purpose and scope of the article are to develop a server that supports map service based on MapBox Vector Tile standard with open source platforms, high-performance optimization, and easy integration for popular platform applications such as WebGIS, MobileGIS, and DesktopGIS. Subjects and methods research using NodeJS platform, PostGIS/PostgreSQL spatial database management system, and based on the open technical standards of MapBox Vector Tile.

The results of the article have built a solution to quickly developing server software that supports mapping services with low-cost as well as take advantage of available open-source technologies for developing a server supporting Vector Tile standard digital map service. This solution will be suitable for small and medium-sized projects, with a limited budget, requiring little implementation time, high data processing efficiency without requiring too large of hardware resources, especially easy to share services with different platforms such as Web, Mobile, and Desktop.

*Keyword: Vector Tile; Tile Map Services; GIS; OpenSource;*