ATLANTIS PRESS

Advances in Computer Science Research, volume 94

International Conference on Big Data, Electronics and Communication Engineering (BDECE 2019)

A Microservice–Based Big Trajectory Data Processing Platform for Multimodal Trip Planning

Jun Na*, Xiaowei Wang, Yuting Xu and Wenzhi Luo
Northeastern University, Shenyang China
*Corresponding author

***Abstract*—Trip planning is an essential function provided by an intelligent transportation system. As many different transport modes have been emerging recently, it becomes a common requirement to mix available transport types to find the most comfort and fastest solutions for getting given destinations. A key challenge for achieving such a desired trip plan is to get quality estimations of all possible traveling services rapidly and accurately. Thanks to the rapid growth of mobile computing, lots of real-time data about traffic status can be collected throw smartphones. Based on such continuously acquired data, we would get quality estimations more accurately and efficiently. Thus, we design and implement a big trajectory data processing platform based on microservices for supporting multimodal trip planning. The paper introduces not only the services and service processes for dealing with the big trajectory data but also the implementation and distributed deployment based on microservices deployed in multiple Dockers. Regarding the loosely-coupled nature of microservices, this platform can provide a flexible way to support trip planning applications consider new available travel modes quickly.**

Keywords—trip planning; big trajectory data processing; multimodal routing; microservices

I. INTRODUCTION

Trip planning is a critical application in intelligent transportation systems [1], whose purpose is to provide a reliable travel plan for users. Currently, we usually plan our trip by map/navigation applications, such as Google map, Baidu map, Gaode map. These apps can calculate and recommend specific routes from the given departure place to the destination. Moreover, they can provide attributes of alternative routes, including the time spent on the whole trip or each part of the trip, the total distance of the trip, the cost of taking a bus or taxi, as well as the total number of transfers. Based on such information, a user can select an appropriate travel route according to their preference conveniently. Unfortunately, most of the existing applications cannot generate routes combining multiple travel modes. For example, combining public transportation with driving or take a taxi. However, traveling by a mixture of available transport types might sometimes result in more comfort and faster solution for passengers to get their destinations. For example, commuters may prefer to drive to regular destinations first and then change to take the subway to save petrol and avoid traffic jams. Therefore, in order to provide more efficient trip plans, applications should be able to provide multimodal trip plans [2, 3].

Shortest path computation is the most common strategy used in route planning. The basis of many state-of-the-art algorithms exploits Dijkstra's algorithm [4]. Usually, the route planning needs an underlying road network which can be modeled as a graph. The nodes represent the junctions in a network, while a link can connect two nodes. Considering composing multiple different travel modes, the trip planning issue turns to be more complicated. In order to supply the necessary information for multimodal trip planning, the network usually consists of many layers [5]. Each layer represents a different transport mode, and interlayer links are representing the transfer from one mode to the other. Moreover, Pareto Sets with multiple criteria are adopted to extend the resulted recommend path set. Currently, most existing products and applications only consider composing walking with public transportation. They cannot create more flexible multimodal routes, such as combining public transportation and self-driving. Furthermore, there are more diverse travel modes emerging, such as shared bicycles, shared motorcycles, shared cars, etc. The issue of how to catch up with the development of available travel modes and provide more efficient and comfortable trip plans is still open and challenging [6-8].

In order to solve the above problems, this paper proposes a microservice-based big trajectory data processing platform for multimodal trip planning. Briefly, we extract three kinds of services, i.e., data management services, trip planning services, and interaction services. For each kind of service, we distinguish data-related services and algorithm-related services. Based on such service classification, we put forward the primary services necessary for creating a multimodal trip plan. Then, all these services constitute a platform to process big trajectory data and provide interfaces for generating routes according to different users' requirements. Thanks to the flexible nature of microservices, future travel modes can be easily considered and extended into current trip planning as a new service. Besides, more efficient algorithms can also be easily adopted in the platform for improving its computational efficiency.

II. SERVICE DESIGN

The microservice architecture is a novel architectural style that is gaining more and more attention [9,10]. Microservices

are lightweight Web services that are smaller and easy to understand, deploy, and scale independently. The most acknowledged definition of microservice is proposed by Fowler and Lewis [11], which describes it as an approach for developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. A microservice is usually applied to model and implement an independent module. Moreover, it can be deployed on single or multiple servers.

In order to identify all necessary trajectory processing services for generating a multimodal trip plan, we first classify services into three different layers, i.e., data service, algorithm service and user interface service. Then, according to the business requirements for generating multimodal trip plans, we extract eleven groups of services as shown in Table I.

A. Data Services

Data services provide all basic CRUD (create, retrieve, update, and delete) operations on data entities from the underlying database systems. As shown in Table I, there are four kinds of information needed for planning multimodal trips, i.e., user information, trajectory data, roadmaps, and travel qualities. Accordingly, we designed four groups of data services to deal with each kind of information. Specially, the roadmap management services and travel quality management services include all kinds of trip modes and each different trip mode has individual roadmap and travel quality data. In other words, we will provide separate roadmap management services and travel quality management services for each kind of trip mode. Then, the resulted platform can be easily extended by deploying more kinds of roadmap management services and travel quality management services.

TABLE I. NECESSARY TRAJECTORY DATA PROCESSING SERVICES IN THE PROPOSED PLATFORM

No.	Service type	Related object	Service function
1	Data service & User interface service	User information	User information management service
2	Data service	Trajectory data	Trajectory data management service
3	Data service	Roadmaps	Roadmap management service
4	Data service	Travel quality information	Travel quality management service
5	Algorithm service	Outlier detection algorithm	Outlier detection services
6	Algorithm service	Trajectory correction algorithm	Trajectory correction services
7	Algorithm service	Roadmap matching algorithm	Roadmap matching services
8	Algorithm service & User interface service	Travel quality predicting algorithm	Travel quality predicting services
9	Algorithm service	Route selection algorithm	Route selection services
10	Algorithm service & User interface service	Trip planning algorithm	Trip planning services
11	User interface service	Trajectory collecting task	Trajectory collecting service

B. Algorithm Services

Algorithm services are designed to encapsulate various algorithms to accommodate dynamic changes in business and technology rapidly and agilely. Considering the algorithms needed for generating a multimodal trip plan, we identified six groups of services, including outlier detection services, trajectory correction services, roadmap matching services, travel quality prediction services, route selection services and trip planning services. Each of these services encapsulates a related algorithm, which guarantees the flexibility of updating algorithms.

C. User Interface Services

User interface services aim to satisfy front-end business requirements. They encapsulate and provide comprehensive services that meet the needs of front-end users. To collect trajectory data from users, we extract the trajectory collecting service, while the user information service, travel quality prediction service, and trip planning service will provide query and planning services to satisfy users' trip planning requirements.

III. SYSTEM ARCHITECTURE

Figure I shows the complete architecture of the proposed microservice-based big trajectory data processing platform. There are three layers, namely, data operation layer, business layer, and interaction layer.

A. Data Operation Layer

The data operation layer provides an interface to access the database, which is used by services that require database operations. It contains all the data services mentioned above and isolates data access from business logic. For services that require some logical operations, we pull data access away from the business logic to improve the maintainability of the service.

B. Business Layer

The business layer contains services that provide the kernel functionalities to process trajectory data for generating multimodal trip plans. There are four simple services which preforms basic tasks in multimodal trip planning.

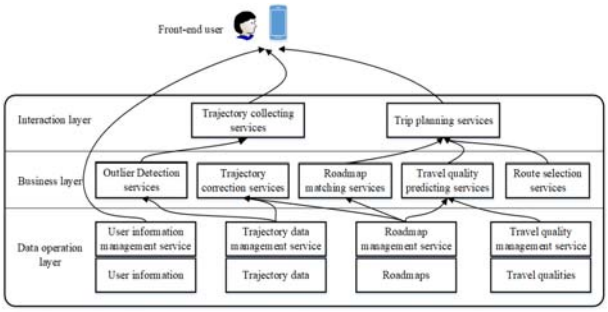

FIGURE I. SYSTEM ARCHITECTURE OF THE PROPOSED BIG TRAJECTORY DATA PROCESSING PLATFORM

1) Outlier detection services: These services encapsulate different outlier detection algorithms. As mentioned above, such a design can support to update algorithms quickly and efficiently. While the trajectory collecting service gets trajectory data from the front-end apps, it will pass the data to the outlier detection service. Based on the support of existing trajectory data, the outlier detection service will detect and mark the abnormal positions and put them to the trajectory dataset through the trajectory data management service.

2) Trajectory correction services: These services encapsulate different trajectory correction algorithms. They clean the current trajectory data period. A trajectory correction service will first get abnormal trajectory data through the trajectory data management service, and then revise these data based on current trajectory data and roadmap. After running the encapsulated trajectory correction algorithm, the trajectory correction service will put all the cleaned data back to the trajectory dataset through the trajectory data management service.

3) Roadmap matching services: These services encapsulate various algorithms to get the roadmap segments by different travel modes. As mentioned above, there are separate roadmaps and relevant roadmap management services. Roadmap matching services will invoke each of these services to select the possible route segments which might be involved in the final trip plans according to given origin and destination.

4) Travel quality predicting services: Silimar roadmaps and roadmap management services, there are also several travel qualities and relevant travel quality management services. Travel quality predicting services aim to encapsulate different prediction algorithms to predict travel time, cost, distance, and other value of travel qualities. Such qualities are necessary for selecting the final group of routes offering to front-end users.

5) Route selection services: Route selection services encapsulate the traditional optimization algorithms, such as Dijkstra's algorithm, Pareto optimization, and other heuristic algorithms to select a group of best routes to satisfy users' requirements. These services are the kernel of generating a multimodal trip plan.

C. Interaction Layer

The interaction layer provides the user with a way to access the services that the system exposes to satisfy front-end requirements. A user can use functions such as account registration, login authentication, and personal information management through user information management service, require trip planning and inquiry travel quality by trip planning services and upload their trajectory data through trajectory collecting service. At the same time, the interaction layer also plays the role of hiding part of the structural/architectural details of the whole platform. Users can only see and use the functions provided directly for them by a group of convenient interfaces, while the complexity of background implementation are shielded and protected.

For example, the trip planning service is a composite service, which encapsulates a processing logic of tasks performed by other services. In order to generate a multimodal trip plan, the trip planning service needs to interact with the roadmap matching service, the travel quality predicting service, and route selection service. When a user provides his/her departure place and destination, the trip planning service will pass this information to the roadmap matching services to find possible paths by different travel mode can be involved in the resulted trip plan. Then, after receiving the path segments from all the roadmap matching services, the trip planning service will then pass these path segments into the travel quality predicting services to get related travel qualities for different path segments. At last, it will invoke the route selection service to generate and recommend a group of trip plans and return the results to the front-end users. However, these complexed workflow is transparent to the front-end users.

IV. IMPLEMENTATION

To implement the proposed service platform, we develop microservices based on the Spring Cloud framework [12] and deploy all the services into Docker containers [13]. Besides, we adopt the OpenStreetMap [14] as the basic roadmap, use HMM to realize roadmap matching [15], and use the shortest path algorithm proposed in [16] to select the optimal routes satisfying users' requirements. At last, we develop an Android APP to collect users' trajectory data and provide trip planning services.

A. Implementation Architecture

To realize all the functions in the form of microservices, we develop all the services based on the Spring Cloud framework and deploy all the services in Docker containers, as shown in Figure II. All the functional services listed in Table I must register into the Spring Cloud Eureka Server. Service invocations are performed through Spring Cloud Zuul, while we adopt Spring Cloud Config Server to perform service configuration. At last, we apply the Docker Compose to realize the composite services by defining and running a complex application based on Docker containers.

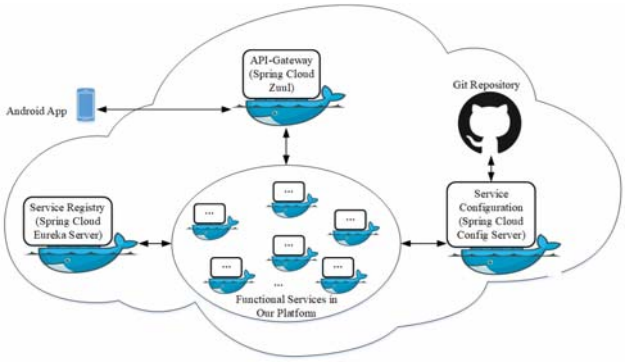

FIGURE II. IMPLEMENTATION ARCHITECTURE BASED ON SPRING CLOUD AND DOCKER

B. *Key Algorithms Implementation*

Trip planning requires a pre-built road network. We adopt the OpenStreeMap, which is an open-source map to capture current map information. There is a large set of data uploaded from volunteers all over the world and can be downloaded freely. We use these data to construct the roadmaps in our platform and developed corresponding services to synchronize changes in maps and provide interfaces to accessing these data.

Based on the open dataset provided by the Microsoft GeoLife project [17], we get the sample trajectory data with travel mode labels. Map matching algorithms require not only GPS data in a trajectory but also the direction and speed of each track point to improve the accuracy of matching. Thus, we first employ each pair of adjacent points to calculate the direction and speed of the original point and then adopt the approaches proposed in [18] to realize the outlier detection and correction.

Hidden Markov based map matching algorithms can consider both the distance between GPS trajectory and roads as well as the road connectivity, which is able to reduce the influence of GPS noise on map matching greatly. Thus, we implement the HMM-based map matching algorithm proposed in [15] and encapsulate it as a service. Similarly, we implement the shortest path computing algorithm proposed in [16] and encapsulate a route selection service.

C. *Android based Application*

We develop a demo app based on the MVP (Model-View-Presenter) architecture in Kotlin. Compare with the traditional MVC architecture, the MVP architecture abstracts the business logic into the Presenter Layer and holds a reference of the View Layer. When responding to the View Layer, we only need to call the reference of the layer, which improves the flexibility of our app.

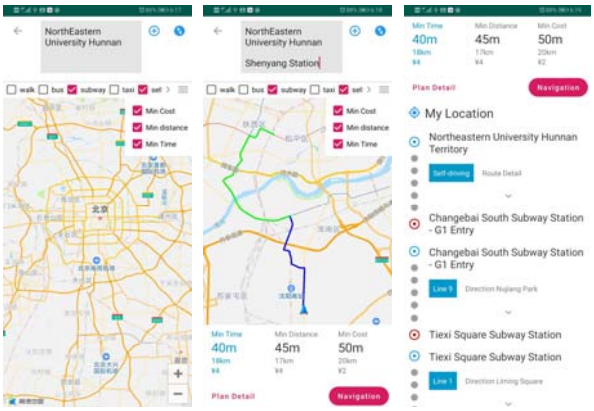

FIGURE III. OUT ANDROID APPLICATION

V. CONCLUSION

This paper describes the design and implementation of a microservice-based big trajectory data processing platform for multimodal trip planning. Based on this platform, a developer can consider future possible travel modes in generating more efficient and suitable trip plans quickly by developing and deploying new services. Moreover, algorithms can also be updated and applied flexibly to provide better services with higher service qualities.

ACKNOWLEDGMENT

This work was supported by the Fundamental Research Funds for the Central Universities of China under Grant No. N181704003

REFERENCES

[1] P. Mrazovic, J. L. Larribapey, and M. Matskin. "Improving mobility in smart cities with intelligent tourist trip planning", *Computer Software and Applications Conference* (2017): 897-907.

[2] M. Braun. "Multi-modal route planning with transfer patterns", Master's thesis, University of Freigurg, 2012.

[3] J. Li, K. Zhou, L. Zhang, and W. Zhang. A multimodal trip planning system with real-time traffic and transit information. Journal of Intelligent Transportation Systems, 2012, 16(2): 60-69.

[4] E. W. Dijkstra. "A note on two problems in connexion with graphs". Numerische Mathematik, Numerische Mathematik, 1959,1(1):269-271.

[5] N. Dimokas, K. Kalogiruou, P. Spanidis, and D. Kehagias. "A mobile application for multimodal trip planning." International Conference on Information Intelligence Systems and Applications (2018): 1-8.

[6] K. H. Lim, J. Chan, S. Karunasekera, and C. Leckie. Tour recommendation and trip planning using location-based social media: a survey[J]. Knowledge and Information Systems, 2019, 60(3): 1247-1275.

[7] X. Kong, M. Li, K. Ma, K. Tian, M. Wang, Z. Ning, and F. Xia. Big trajectory data: a survey of applications and services[J]. IEEE Access, 2018: 58295-58306.

[8] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang. Big data analytics in intelligent transporation systems: a survy. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(1): 383-398.

[9] M. Garriga. Towards a taxonomy of microservices architectures. *International Conference on Software Engineering* (2017): 203-218.

[10] P. Jamshidi, C. Pahl, N. C Mendonca, J. Lewis, and S. Tilkov. Microservices: the journey so far and challenges ahead. IEEE Software, 2018, 35(3): 24-35.

[11] J. Lewis and M. Fowler. Microservices: a definition of this new architecturural term. 2014. https://martinfowler.com/articles/microservices.html

[12] Spring Cloud. https://spring.io/projects/spring-cloud

[13] Docker. https://www.docker.com

[14] OpenStreetMap. https://www.openstreetmap.org

[15] P. E. Newson and J. Krumm. Hidden markov map matching through noise and sparseness. Advances in Geographic Information Systems, 2009: 336-343.

[16] O. Dib, M. Manier, and A. Caminada. Memetic algorithm for computing shortest paths in multimodal transportation networks. Transporation Research Procedia, 2015: 745-755.

[17] GeoLife GPS Trajectories Dataset. https://www.microsoft.com/en-us/download/details.aspx?id=52367

[18] Y. Zheng. Trajectory data mining: an overview. ACM Transactions on Intelligent Systems and Technolgoy, 2015, 6(3):1-41.