Multi-Cloud-Based Transportation Recommendation System

Anirudh Sharma
sharmani@iu.edu

Harshini M N
hmn@iu.edu

Kamalesh Kumar MG
kmandak@iu.edu

Project Goal

The project aims to develop a multi-cloud-based transportation recommendation system that helps users find the nearest and best modes of transportation between two cities. This system will integrate resources from multiple cloud providers such as GCP, Azure and AWS to ensure interoperability, reliability, scalability, and redundancy.

Objectives:

1. To design and implement a transportation recommendation system.
2. To integrate resources from multiple cloud providers for redundancy and scalability.
3. To provide users with real-time transportation options based on their preferences.
4. To optimize route recommendations for time, and environmental impact.
5. To evaluate the system's performance, scalability, and user satisfaction.

Related Work

In the realm of multi-cloud technology and its applications, several noteworthy studies provide valuable insights into the challenges and solutions associated with cloud-based intelligent transport systems, making them pivotal references for our project.

1. Multi-Cloud Technology: Challenges and Solutions
Hamza Ali Imran, Usama Latif, Ataul Aziz Ikram, Maryam Ehsan, and Ahmed Jamal Ikram, in their paper "Multi-Cloud: A Comprehensive Review" published by IEEE, delve into the complexities of multi-cloud technology. Their exploration of transitioning between different cloud services and addressing parameters like time, cost, performance, and security offers crucial foundational knowledge. This understanding is invaluable as we design a Multi-Cloud-Based Transportation Recommendation System, ensuring seamless integration and optimal performance across diverse cloud platforms [1].

2. Managing Resources in Multi-Cloud Environments
Victor Ion Munteanu, Călin Şandru, and Dana Petcu's research on "Multi-Cloud Resource Management: Cloud Service Interfacing" provides essential insights into managing resources in multi-cloud environments. Their work on mOSAIC, an open-source Platform-as-a-Service, illuminates the challenges of interoperability and complexity associated with diverse cloud

service providers. By addressing these challenges, we can develop a recommendation system that can harness the power of multiple cloud platforms efficiently [2].

3. Semantic-Interoperability in Multi-Cloud Platform Management
The study conducted by Eleni Kamateri, Nikolaos Loutas, Dimitris Zeginis, James Ahtes, Francesco D'Andria, Stefano Bocconi, Panagiotis Gouvas, Giannis Ledakis, Franco Ravagli, Oleksandr Lobunets, and Konstantinos A. Tarabanis on "Cloud4SOA: A Semantic-Interoperability PaaS Solution for Multi-cloud Platform Management and Portability" is particularly relevant to our project. Their focus on semantic technologies and seamless communication between diverse cloud services aligns with our goal of enhancing the Multi-Cloud-Based Transportation Recommendation System's interoperability and data exchange capabilities [3].

4. Cloud Technology in Transportation Systems
The integration of cloud technology into transportation systems is a topic explored in-depth in a study titled "Cloud Based Intelligent Transport System." This research emphasizes how cloud computing can transform transportation networks by enabling smarter and more efficient systems. By understanding these advancements, we can incorporate intelligent features into our recommendation system, enhancing the overall travel experience for users [4].

5. Data Protection, Privacy, and Open Research Challenges in Cloud Computing
Junaid Hassan, Danish Shehzad, Usman Habib, Muhammad Umar Aftab, Muhammad Ahmad, Ramil Kuleev, and Manuel Mazzara's systematic literature review on "The Rise of Cloud Computing: Data Protection, Privacy, and Open Research Challenges" highlights the critical aspects of data protection and privacy within cloud computing. This review informs our project about the challenges associated with safeguarding user data, ensuring privacy, and addressing ongoing research challenges, ensuring the security and integrity of our Multi-Cloud-Based Transportation Recommendation System [5].

By drawing upon these diverse studies, our project gains a comprehensive understanding of the multi-cloud landscape, enabling us to design a robust and efficient Multi-Cloud-Based Transportation Recommendation System. Each of these sources contributes valuable insights, shaping the foundation of our research and development efforts.

Gap Analysis

One of the main challenges in developing a multi-cloud-based transportation recommendation system is integrating resources from multiple cloud providers. This can be complex and expensive, as it requires developing interfaces and communication protocols that enable data and services to flow seamlessly between providers. Additionally, it is important to carefully consider data formats, APIs, and authentication mechanisms to ensure that services from different cloud providers can work together.

- **Lack of unified recommendation systems**

  Various cloud-based platforms offer transportation-related services. For instance, Google Cloud Platform (GCP) offers APIs for route planning and traffic prediction, while AWS provides geospatial data services. Azure offers services related to IoT and data analytics for transportation. However, there is limited research on creating a unified recommendation system that leverages the strengths of multiple cloud providers. This is a gap in the current state of the art, as a unified recommendation system could provide a number of benefits, such as:

  - **Increased performance and scalability**: By leveraging the resources of multiple cloud providers, a unified recommendation system can handle a large number of users and requests.
  - **Improved reliability**: If one cloud provider experiences downtime or issues, the system can seamlessly switch to another provider to maintain service availability.
  - **Reduced costs**: By using a pay-as-you-go model, a unified recommendation system can help to reduce costs.

  Ensuring that services from different cloud providers can work together is a significant challenge in multi-cloud integration. The project focuses on creating interfaces and communication protocols that enable data and services to seamlessly flow between providers. This requires careful consideration of data formats, APIs, and authentication mechanisms.

  Another gap is that existing transportation recommendation systems often focus on optimizing for travel time and cost. However, there is a growing demand for transportation recommendation systems that can also optimize for other factors, such as environmental impact, accessibility, and user preferences

- **Limited support for user preferences**

  Another gap in existing transportation recommendation systems is their limited support for user preferences. Existing systems often focus on optimizing for travel time and cost, but there is a growing demand for systems that can also optimize for other factors, such as environmental impact, accessibility, and user preferences. For example, a user may prefer to take public transportation over driving if it is more environmentally friendly, or they may prefer to take a route that is wheelchair accessible. The proposed project will focus on :

- **User-centric**: The system will consider user preferences, such as travel time, cost, environmental impact, and accessibility, when recommending transportation options.
- **Data-driven**: The system will be powered by a large amount of transportation data, including real-time traffic data, public transportation schedules, and ride-hailing service availability data.

Proposed Tasks

- To design a user interface (Web or Mobile App) where users can input their origin and destination cities. They can also specify preferences like travel time, or environmental impact.
- Design algorithms that consider user preferences, such as travel time, cost, environmental impact, and accessibility when recommending transportation options.
- Identify the strengths of each cloud provider, such as GCP's route planning APIs and AWS's geospatial data services, and design a strategy for efficient resource utilization.
- Store transportation data, user preferences, and recommendations. Implement data synchronization between cloud providers for consistency.
- Create a unified API layer that abstracts the variations in APIs offered by different cloud providers.
- Conduct extensive testing to ensure that data and services can seamlessly flow between cloud providers.

Microservice Architecture

Our project employs a microservices architecture, dividing the system into modular and independently deployable components. This architecture enhances scalability, maintainability, and flexibility, allowing each microservice to perform specific functions.

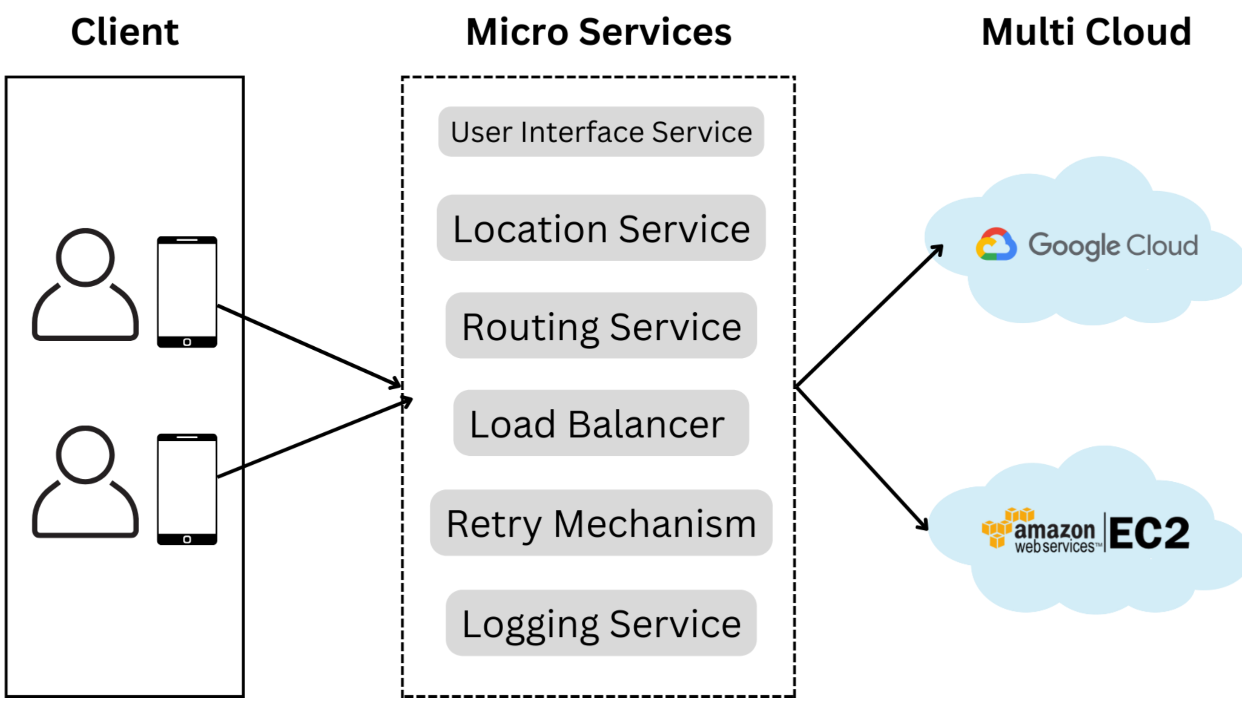

Fig.1 - Microservice architecture with Multi cloud

Microservices Components

- User Interface Service:
  The user interface for the transportation recommendation system employs simple Javascript and HTML for the frontend. The User Interface Service enables users to input origin and destination details, specify preferences, and interact with the system seamlessly.

- Location Service:
  Retrieves the user's geographical coordinates based on their IP address. This service uses Flask, IPStack API. The Location Service uses the IPStack API to determine the user's location, contributing to personalized transportation recommendations.

- Load Balance Service:
  Balances the load and selects the appropriate cloud provider for transportation recommendations. Technologies used are Flask, Google Cloud Monitoring API, AWS CloudWatch. The Load Balance Service optimizes resource utilization by considering factors such as CPU utilization and location, ensuring efficient recommendation delivery.

- Routing Service:

Interacts with the chosen cloud provider to fetch real-time transportation options. Technologies include Flask, Requests library. The Routing Service communicates with cloud providers like Google Cloud Platform and AWS to obtain up-to-date transportation recommendations.

- Inter-Microservices Communication:
  Microservices communicate through HTTP requests, fostering a decoupled and distributed system. For example, the User Interface Service interacts with the Location Service to obtain user coordinates, and the Load Balance Service orchestrates the selection of the cloud provider based on system load.

- Scalability and Redundancy:
  The microservices architecture facilitates scalability as each component can be independently scaled based on demand. Redundancy is achieved by distributing functionalities across multiple microservices, ensuring system resilience.

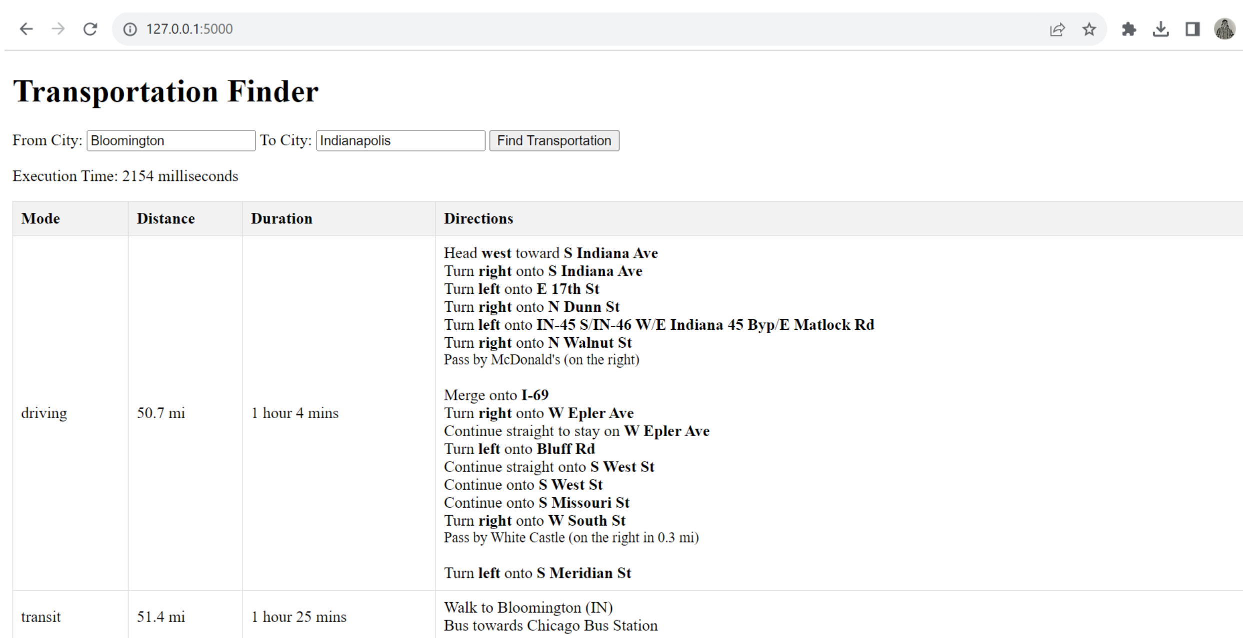

Fig.2 - Screenshot of user interface service

Challenges and Solutions

While the microservices architecture brings numerous benefits, challenges such as inter-service communication and data consistency must be addressed. We implement RESTful APIs for seamless communication and employ strategies like eventual consistency to handle data synchronization challenges.

Progress on Microservices Implementation

- Location Service Integration
  We have successfully integrated the Location Service, utilizing the IPStack API to fetch accurate user coordinates based on their IP address.
- Load Balance Service Enhancement
  The Load Balance Service has been enhanced to consider CPU utilization metrics from cloud providers, ensuring intelligent load distribution.
- Routing Service Development
  The Routing Service is under active development, establishing connections with cloud providers and implementing robust mechanisms for real-time transportation data retrieval.
- Testing and Optimization

Extensive testing is underway to validate the reliability and performance of each microservice. Optimization strategies are implemented to enhance response times and resource utilization.

```
 * Running on http://127.0.0.1:5000
2023-12-08 20:29:52,821 - INFO - ␛[33mPress CTRL+C to quit␛[0m
2023-12-08 20:29:52,824 - INFO -  * Restarting with stat
2023-12-08 20:29:53,953 - WARNING -  * Debugger is active!
2023-12-08 20:29:53,957 - INFO -  * Debugger PIN: 313-738-329
2023-12-08 20:29:56,213 - INFO - 127.0.0.1 - - [08/Dec/2023 20:29:56] "GET / HTTP/1.1" 200 -
2023-12-08 20:29:56,834 - INFO - 127.0.0.1 - - [08/Dec/2023 20:29:56] "␛[33mGET /favicon.ico HTTP/1.1␛[0m" 404 -
2023-12-08 20:30:05,586 - DEBUG - Client IP: [illegible]
2023-12-08 20:30:05,592 - DEBUG - Starting new HTTP connection (1): api.ipstack.com:80
2023-12-08 20:30:05,737 - DEBUG - http://api.ipstack.com:80 "GET /[illegible]?access_key=[illegible] HTTP/1.1" 200 None
2023-12-08 20:30:05,739 - DEBUG - IP Stack Response: {'ip': '[illegible]', 'type': 'ipv4', 'continent_code': 'EU', 'continent_name': 'Europe',
'country_code': 'SE', 'country_name': 'Sweden', 'region_code': 'E', 'region_name': '␛sterg␛tland', 'city': 'Link␛ping', 'zip': '582 22', 'latitude':
58.41175079345703, 'longitude': 15.615289688110352, 'location': {'geoname_id': 2694762, 'capital': 'Stockholm', 'languages': [{'code': 'sv', 'name':
'Swedish', 'native': 'Svenska'}], 'country_flag': 'https://assets.ipstack.com/flags/se.svg', 'country_flag_emoji': '\U0001f1f8\U0001f1ea',
'country_flag_emoji_unicode': 'U+1F1F8 U+1F1EA', 'calling_code': '46', 'is_eu': True}}
2023-12-08 20:30:05,739 - DEBUG - Client Coordinates: (58.41175079345703, 15.615289688110352)
2023-12-08 20:30:05,739 - DEBUG - Ordered Cloud Providers: ['GCP', 'AWS']
2023-12-08 20:30:05,739 - DEBUG - Chosen Cloud Provider: GCP
2023-12-08 20:30:05,739 - DEBUG - Request sent to GCP provider
2023-12-08 20:30:05,742 - DEBUG - Starting new HTTPS connection (1): tribal-bay-407302.oa.r.appspot.com:443
2023-12-08 20:30:07,720 - DEBUG - https://tribal-bay-407302.oa.r.appspot.com:443 "GET /find_best_transportation?from_city=Bloomington&to_city=Indianapolis
HTTP/1.1" 200 None
```

Fig.3 - Screenshot of request served through GCP

```
2023-12-08 10:47:53,538 - INFO -  * Detected change in 'D:\\IU\\Academics\\Semester 3\\ECC\\Final Project\\poc\\multicloud\\app.py', reloading
2023-12-08 10:47:53,651 - INFO -  * Restarting with stat
2023-12-08 10:47:55,072 - WARNING -  * Debugger is active!
2023-12-08 10:47:55,075 - INFO -  * Debugger PIN: 313-738-329
2023-12-08 10:48:03,540 - INFO - 127.0.0.1 - - [08/Dec/2023 10:48:03] "GET / HTTP/1.1" 200 -
2023-12-08 10:48:11,342 - DEBUG - Client IP: [illegible]
2023-12-08 10:48:11,346 - DEBUG - Starting new HTTP connection (1): api.ipstack.com:80
2023-12-08 10:48:11,446 - DEBUG - http://api.ipstack.com:80 "GET /[illegible]?access_key=[illegible] HTTP/1.1" 200 None
2023-12-08 10:48:11,447 - DEBUG - IP Stack Response: {'ip': '[illegible]', 'type': 'ipv4', 'continent_code': 'NA', 'continent_name': 'North America',
'country_code': 'US', 'country_name': 'United States', 'region_code': 'CA', 'region_name': 'California', 'city': 'Burlingame', 'zip': '94010', 'latitude':
37.56782913208008, 'longitude': -122.36945343017578, 'location': {'geoname_id': 5331920, 'capital': 'Washington D.C.', 'languages': [{'code': 'en', 'name':
'English', 'native': 'English'}], 'country_flag': 'https://assets.ipstack.com/flags/us.svg', 'country_flag_emoji': '\U0001f1fa\U0001f1f8',
'country_flag_emoji_unicode': 'U+1F1FA U+1F1F8', 'calling_code': '1', 'is_eu': False}}
2023-12-08 10:48:11,447 - DEBUG - Client Coordinates: ([illegible], [illegible])
2023-12-08 10:48:11,447 - DEBUG - Ordered Cloud Providers: ['AWS', 'GCP']
2023-12-08 10:48:11,447 - DEBUG - Chosen Cloud Provider: AWS
2023-12-08 10:48:11,447 - DEBUG - Request sent to AWS provider
2023-12-08 10:48:11,449 - DEBUG - Starting new HTTP connection (1): 3.144.72.249:80
```

Fig.4 - Screenshot of request served through AWS

Conclusion

The adoption of a microservices architecture significantly contributes to the success of our Multi-Cloud-Based Transportation Recommendation System. This modular approach streamlines development, fosters agility, and positions our system for future enhancements and integrations.

Key Considerations and Model Evaluation

Cloud Flexibility: Our system is currently deployed on Google Cloud Platform (GCP) and Amazon Web Services (AWS). The configurations of the Flask applications are designed to be adaptable, allowing seamless expansion to other cloud providers. This flexibility ensures that our system remains scalable and can harness the strengths of diverse cloud environments.

Multi-Cloud Strategy: Future iterations of our project will leverage a multi-cloud strategy, distributing different services across various cloud providers. This approach mitigates the risk of vendor lock-in and enhances system resilience. In the event of service interruptions from one cloud provider, other services can continue to function independently, preventing a complete system shutdown.

Supply Chain Focus: Our product is purpose-built with supply chain companies in mind. The transportation recommendation system aims to address the specific needs of supply chain logistics, providing efficient, real-time recommendations that optimize routes, reduce costs, and enhance overall supply chain management.

Future Roadmap

Our vision for the future involves hosting different services in diverse cloud environments. This strategic multi-cloud approach aligns with our commitment to providing a robust and reliable transportation recommendation system. By avoiding reliance on a single cloud provider, we ensure the continuous operation of critical services, fostering long-term sustainability and minimizing potential disruptions.

In summary, our Transportation Recommendation System shows our dedication to being creative, flexible, and solving problems in the supply chain field. As we continue, adding more cloud options and offering new services will help us become even better at providing smart transportation solutions.

Project Links:

Amazon Web Services URL: http://3.144.72.249
Google cloud platform URL: https://tribal-bay-407302.oa.r.appspot.com
Git URL: https://github.iu.edu/hmn/ENGR516-MultiCloud

References:

[1] Hamza Ali Imran, Usama Latif, Ataul Aziz Ikram, Maryam Ehsan, and Ahmed Jamal Ikram. "Multi-Cloud: A Comprehensive Review." IEEE. https://ieeexplore.ieee.org/abstract/document/9318176

[2] Victor Ion Munteanu, Călin Şandru, and Dana Petcu. "Multi-Cloud Resource Management: Cloud Service Interfacing." Journal of Cloud Computing. https://link.springer.com/article/10.1186/2192-113X-3-3

[3] Eleni Kamateri, Nikolaos Loutas, Dimitris Zeginis, James Ahtes, Francesco D'Andria, Stefano Bocconi, Panagiotis Gouvas, Giannis Ledakis, Franco Ravagli, Oleksandr Lobunets, and Konstantinos A. Tarabanis. "Cloud4SOA: A Semantic-Interoperability PaaS Solution for Multi-cloud Platform Management and Portability." https://link.springer.com/chapter/10.1007/978-3-642-40651-5_6

[4] Cloud Based Intelligent Transport System. https://www.sciencedirect.com/science/article/pii/S1877050915005621

[5] Junaid Hassan, Danish Shehzad, Usman Habib, Muhammad Umar Aftab, Muhammad Ahmad, Ramil Kuleev, and Manuel Mazzara. "The Rise of Cloud Computing: Data Protection, Privacy, and Open Research Challenges—A Systematic Literature Review (SLR)." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9197654/