Group Task (Module 2)

Big Data Process Mapping – Google Maps

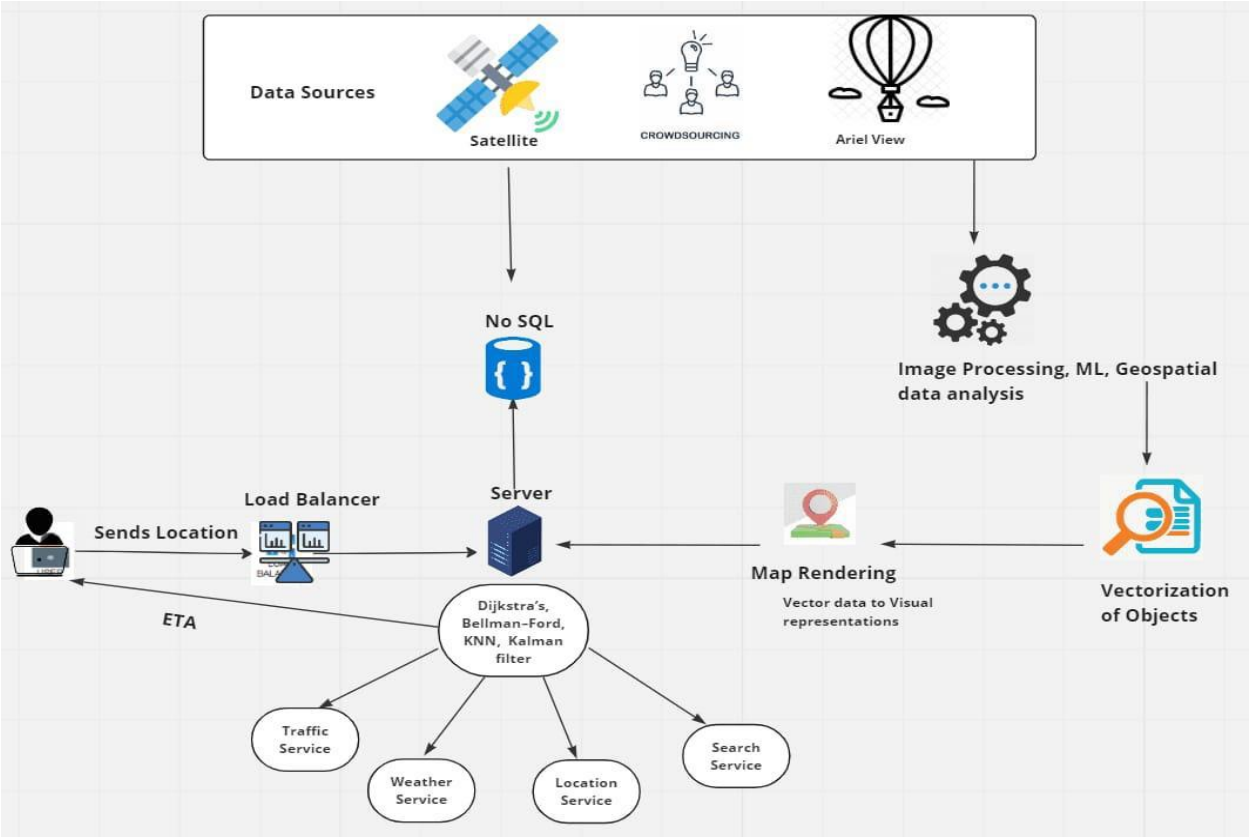
1. Introduction

Google Maps is one of the most powerful real-world applications of Big Data and distributed systems. What appears to users as a simple navigation app is actually a large-scale geo-spatial data platform operating globally. It processes billions of location updates daily and serves millions of routing requests per minute.

The system must handle continuous real-time streaming data, store massive historical datasets, apply predictive machine learning models, and compute optimized routes — all within milliseconds. This makes Google Maps an ideal example of how modern Big Data systems combine scalability, speed, intelligence, and reliability.

This report explains the complete end-to-end data flow, architecture, storage strategy, processing mechanisms, and Big Data characteristics of Google Maps.

2. System Architecture Overview



Google Maps follows a hybrid architecture inspired by the Lambda model, integrating both real-time and batch processing.

The architecture consists of three main layers:

2.1 Speed Layer (Real-Time Processing)

This layer handles live GPS streams and traffic updates. It processes congestion levels and updates road conditions within seconds.

2.2 Batch Layer (Historical Processing)

The batch layer analyzes stored historical data. It improves machine learning models and rebuilds road network indexes periodically.

2.3 Serving Layer (Response Layer)

The serving layer delivers route results and traffic updates to users through low-latency APIs.

This layered structure ensures both instant updates and long-term predictive accuracy.

3. End-to-End Data Flow

The complete system pipeline operates continuously:

1. Mobile devices generate GPS coordinates, speed, and timestamps.
2. Data is transmitted securely to regional edge servers.
3. A distributed Pub/Sub system organizes data geographically.
4. Stream processing engines analyze traffic in real time.
5. Updated traffic metrics are stored in hot databases.
6. Machine learning models predict future traffic conditions.
7. Routing algorithms compute the fastest route.
8. The API gateway sends the final response back to the user.

The entire process is optimized for minimal latency.

4. Data Sources

Google Maps integrates multiple data sources.

4.1 Real-Time Streaming Data

- GPS coordinates
- Speed and movement direction
- Timestamp
- User-reported incidents

This data is high in velocity and arrives continuously.

4.2 Structured Batch Data

- Road network graphs
- Speed limits
- GIS datasets
- Construction and road closure updates
- Points of interest

These datasets form the foundation of routing decisions.

5. Storage Architecture

Google Maps uses a dual storage strategy.

5.1 Hot Storage

- Stores real-time traffic metrics
- Provides fast read/write operations
- Lookup latency below 10 milliseconds
- Used for immediate route computation

5.2 Cold Storage

- Stores historical traffic data
- Implemented using distributed file systems
- Used for analytics and ML model training

This separation ensures both speed and scalability.

6. Real-Time Stream Processing

The speed layer performs several operations:

6.1 Map Matching

Raw GPS signals are aligned with actual road segments using probabilistic models.

6.2 Traffic Aggregation

Using sliding time windows (e.g., 60 seconds), average road speeds are calculated:

Average Speed = Total Speed / Number of Vehicles

6.3 Congestion Detection

Congestion Score = 1 − (Current Speed / Speed Limit)

6.4 Latency Target

The full processing cycle remains under 5 seconds.

7. Machine Learning Layer

Machine learning improves accuracy and prediction.

7.1 Traffic Prediction

Models analyze:

- Historical speeds
- Time of day
- Day of week
- Weather conditions
- Road type

The output predicts future traffic speeds.

7.2 ETA Prediction

Regression models estimate arrival time using route length, predicted congestion, and historical travel patterns.

8. Routing Engine

The road network is represented as a directed weighted graph:

- Nodes = intersections
- Edges = road segments
- Edge weight = predicted travel time

Algorithms used:

- A*
- Optimized Dijkstra

The engine considers:

- Road closures
- Turn restrictions
- Vehicle types

Route calculations are completed in milliseconds.

9. Serving Layer

The serving layer ensures smooth global performance.

- Stateless microservices
- Containerized infrastructure
- Auto-scaling clusters
- Global data centers

Performance target:

- 95% of responses under 200 milliseconds

10. Big Data Characteristics (5 V's)

Dimension	Implementation in Google Maps
Volume	Stores petabytes of global map and traffic data.
Velocity	Processes continuous GPS streams from millions of users.
Variety	Handles GPS data, GIS maps, road graphs, text reports, and imagery.
Veracity	Applies filtering and validation to remove inaccurate or noisy data.

Dimension	Implementation in Google Maps
Value	Provides optimized routes and accurate ETA predictions.

11. Conclusion

Google Maps is a large-scale distributed Big Data system that integrates streaming computation, distributed storage, machine learning, and graph algorithms into one unified architecture.

It continuously ingests geo-spatial data, processes it in real time, predicts traffic conditions, and delivers optimized navigation routes globally.

Google Maps demonstrates how modern Big Data technologies transform raw data into intelligent services that improve everyday life.