Final Project - The Analysis Report

Esgar E. Jimenez

University of Advancing Technology

MS549: Data Structures & Testing

Abstract

For this project, the goal was to develop and analyze an event-driven ride-sharing simulation, designed to evaluate system performance and rider experience under dynamic demand. The ride-share simulation aims to be similar to Uber and Lyft as it would contain a city map, a fleet of vehicles, a random number of riders, and an estimation of average wait times, arrival times, and overall trip times. The simulation is to integrate multiple advanced data structures and algorithms, such as a graph-based city map, Dijkstra's shortest path routing, and a quadtree spatial index for driver matching. One test I ran was to simulate 2000-time units(for the sake of this paper, a time unit will equal a minute), which means the simulation ran for $\cong 33\ hours$ (simulated, not actual). The car fleet was a total of 50 cars, and rider requests arrived on average every 30-time units, which would be every ≅ 30 minutes (simulated). Across this simulated period, the system completed a total of 392 trips. The rider experience was an average wait time of 0.03-time units, which is equal to ≅ 2 seconds (simulated). This indicates a near-instantaneous pickup and an extremely high service reliability. The average trip duration was 2.73-time units, which is equivalent to≅ 2 minutes 44 seconds (simulated). This reflects the small map scale and the short travel distances. The driver utilization is very though as it was only at 1.07%, this shows that the car fleet size was much greater than necessary, resulting in many vehicles being idle. Overall, the system design delivered an excellent rider experience, but it is inefficient in the allocation of resources. Future work would point to scaling the map size and working with a dynamic car fleet size to improve vehicle utilization.

Keywords: Key Performance Indicators (KPI),

Introduction

Ride-sharing services like Uber and Lyft must balance their riders' satisfaction with operational efficiency. A significant challenge in this problem is the efficiency of matching riders to available cars while minimizing both wait times and idle driver capacity. This project aims to model a simplified ride-sharing system using data structures and algorithms.

The key questions guiding this simulation are:

1. How efficiently does a fleet serve riders under defined arrival rates?
2. What is the average rider experience in terms of wait times and trip durations?
3. How effectively are drivers utilized relative to the total available fleet hours?

To investigate these questions, the simulation was designed using an event-driven engine supported by a graph-based city map for routing and spatial indexing.

Methodology

This section outlines the simulation's architecture, including the core data structures, algorithms, and parameters used to model the ride-sharing system. The simulator was built around a discrete-event simulation engine powered by a priority queue (min-heap via Python's heapq). Events such as rider requests, car arrivals, and drop-offs were all processed in chronological order, ensuring accurate sequencing of the actions. A graph data structure was used to store the city map in adjacency list format, with each node tied to specific x, y coordinates. This allowed for the integration of Dijkstra's algorithm for accurate shortest-path routing between riders and drivers. A quadtree spatial index was used to accelerate nearest-neighbor car searches and to improve performance compared to a brute-force approach. The brute-force was used as a backup or failsafe in the event the quadtree failed for application durability. Cars and riders were modeled as objects with state attributes. Cars tracked availability, location, and assigned riders. While the riders tracked status and timestamps for requests, pickup, and drop-off events. This system was critical for logging each trip's wait time, duration, and

completion. For analysis, the simulation was executed for 2000-time units ($\cong 30\ minutes$), with a fleet of 50 cars seeded across the map. Rider requests were generated dynamically at random intervals, with a mean arrival time of 30-time units ($\cong 30\ minutes$). These parameters resulted in the data set that is analyzed in the following section. With these components in place, the simulation was executed to generate the measurable results that follow.

Results

This results section presents the objective outcome of the simulation. It includes key performance indicators (KPI) and an integrated visualization produced by the system once the simulation is completed. The resulting visualization of the data also creates a histogram of wait times. Once the simulation was run and concluded, the following metrics were produced:

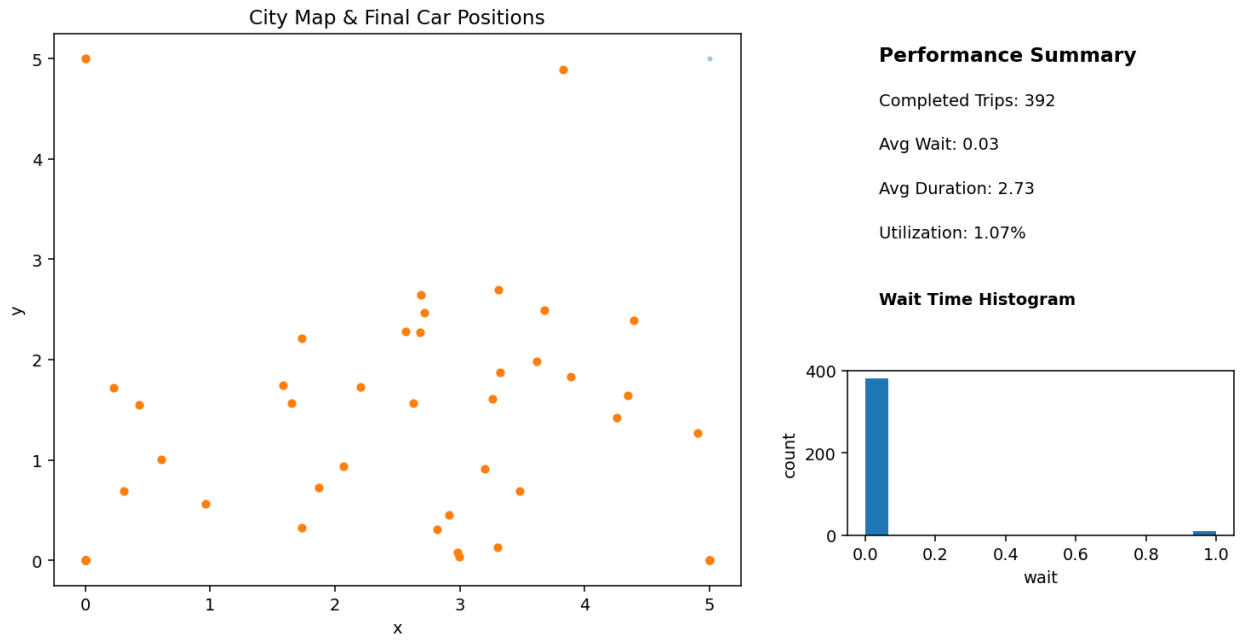
Table 1:

Table 1 Shows the Key Performance Indicators (KPIs) from a simulation run 2000-time units (33 hours), 50 cars, rider requests every 30-time units (30 minutes).

Metric	**Value**
Completed Trips	392
Average Rider Wait Time	0.03
Average Trip Duration	2.73
Driver Utilization (%)	1.07

In addition to the above metrics, the simulation generated an integrated visualization of the data in a file named simulation_summary.png as seen in the following image:

Figure 1 Shows the KPIs in a visualization of the data as a PNG file.



Combining the final map state of the cars with textual KPI annotations. This visualization served as a graphical summary of the simulation activity and results. While the raw metrics provide a snapshot of the system performance, they do require further analysis to understand their implications for efficiency and rider experience.

Discussion

This discussion section is to interpret the results in the section above. We will examine trade-offs, limitations, and hypotheses about how the system performance might change under different conditions. The simulation's low average wait time (0.03-time units $\approx 2\ seconds$) suggests a highly responsive system where riders are served almost immediately upon making a request. This efficiency highlights the effectiveness of the event-driven engine design and the integrated use of pathfinding and spatial indexing in this small map simulation with abundant resources (a car fleet being 50 for a small map). From the rider's perspective, the experience was exceptionally positive based on the simulation results. The issue comes through in utilization; the driver utilization is at a low 1.07%. This means that the fleet was underutilized relative to the demand. This imbalance shows that while the system excels in

responsiveness, the number of cars deployed was excessive compared to the rider arrival frequency. In reality, this would represent wasted resources and potential financial inefficiency.

A clear trade-off shows itself, adding more cars reduces rider wait time but lowers driver utilization. This means that if we reduce the number of cars in the fleet, that will increase the utilization but may lengthen riders' wait times. For example, halving the number of cars to 25 might significantly increase utilization while still keeping wait times at a reasonable length. Other changes, such as larger map graphs or increased distance between paths, could also significantly impact both metrics. Limitations uncovered for the current model include the absence of traffic congestion, real-world delays, driver breaks, multi-rider pooling, vehicle breakdowns, and long-distance trips. Also, trips were modeled on simplified Manhattan distances rather than more nuanced road conditions.

A reasonable hypothesis is that reducing the fleet size or increasing rider request frequency would raise utilization, while maintaining acceptable wait times. Thereby creating a more balanced and realistic ride-sharing environment. Based on the limitations of the simulation, this is a valid hypothesis, but increasing the size of the map could also impact this simulation. Taking all this into account, these results highlight both the strengths, weaknesses, and limitations of the current model, offering insight into opportunities for enhancements and innovation.

Conclusion. In conclusion, this project successfully demonstrated the design and execution of an event-driven ride-sharing simulator using advanced data structures. The results revealed that the system produced a near-instant rider service, but at the cost of low vehicle utilization due to an oversized fleet for the map size and rider request frequency. Future work on this project should focus on experimenting with bigger graph-based maps, different fleet sizes, higher demand rates, and figuring out how to implement real-world delays and other limitations identified.