INTERNATIONAL JOURNAL OF MERGING TECHNOLOGY AND ADVANCED RESEARCH IN COMPUTING

ISSN: 2320-1363

Cloud Based Real Time City Scale Taxi Ridesharing Using GAE

S.Srinivasan[1], K.Poovazhaki[2], A.Sudhakar[3]

[1] M.E Scholar, Dept of CSE, Bharathidasan Engineering College, Vellore, Tamilnadu, India.

[2]Assistant Professor, Dept of CSE, Bharathidasan Engineering College, Vellore, Tamilnadu, India.

[3]HOD and Professor, Dept of CSE, Bharathidasan Engineering College, Vellore, Tamilnadu, India.

Abstract

We proposed and developed a taxi ridesharing system that accepts taxi passenger's real time ride requests sent from smartphones and schedules proper taxis to pick up them via ridesharing, subject to time, capacity, and monetary constraints. The monetary constraints provide incentives for both passengers and taxi drivers. While such a system is of significant social and environmental benefit, e.g., saving energy consumption and satisfying people's commute, we devise mobile cloud architecture based taxi ridesharing system. The system achieves the Digital India and reduces the environmental pollution. Taxi riders and taxi drivers use the taxi sharing service provided by the system via a smart phone App.The Cloud first finds candidate taxis quickly for a taxi ride request using a taxi searching algorithm supported by a spatio-temporal index.GPS takes care of the current location for the users. The cloud storage is used to store the information of the user and synced with Google App Engine. A scheduling process is then performed in the cloud to select a taxi that satisfies the request with minimum increase in travel distance. The complete application works on the basis of SaaS. Our proposed system demonstrated its efficiency, effectiveness and scalability.

Keywords: mobile cloud architecture, taxi ridesharing system, taxi searching algorithm, GPS, SaaS, spatio-temporal index, Google App Engine.

ISSN: 2320-1363

1. INTRODUCTION

Taxi is an important transportation mode between public and private transportations, delivering millions of passengers to different locations in urban areas[9]. However, taxi demands are usually much higher than the number of taxis in peak hours of major cities, resulting in that many people spend a long time on roadsides before getting a taxi.

Increasing the number of taxis seems an obvious solution .But it brings some negative effects, e.g., causing additional traffic on the road surface and more energy consumption and decreasing taxi driver's income (considering that demands of taxis would be lower than number of taxis during off-peak hours).we propose a taxi sharing system that accepts taxi passengers real time ride requests sent from smartphones and schedules proper taxis to pick up them via taxi sharing with time, capacity, and monetary constraints (the monetary constraints guarantee that passengers pay less and drivers earn more compared with no taxi-sharing is used)[16]. Our system saves energy consumption and eases traffic congestion while enhancing the capacity of commuting by taxis. Meanwhile, it reduces the taxi fare of taxi riders and increases the profit of taxi drivers. system based on the mobile cloud architecture, which enables real time taxi sharing in a practical setting. In the system, taxi drivers independently determine when to join and leave the service using an App installed on their smartphones. Passengers submit real time ride requests using the same App (if they are willing to share the ride with others)[13]. Each ride request consists of the origin and destination of the trip, time windows constraining when the passengers want to be picked up and dropped off. On receiving a new request, the Cloud will first search for the taxi which minimizes the travel distance increased for the ride request and satisfies both the new request and the trips of existing passengers who are already assigned to the taxi, subject to time, capacity, and monetary constraints. Then the existing passengers assigned to the taxi will be inquired by the cloud whether they agree to pick up the new passenger given the possible decrease in fare and increase in travel time. Only with a unanimous agreement, the updated schedules will be then given to the corresponding taxi drivers and passengers.constraints which include time windows, capacity and monetary constraints for taxi trips. In addition, our work proposes efficient searching and scheduling algorithms that are capable of allocating the "right" taxi among number of taxis for a query in milliseconds.

We proposed and developed a taxi ride sharing system using the mobile cloud architecture. The cloud integrates multiple important components including taxi indexing, searching, and scheduling. Specifically, we propose a spatio-temporal indexing structure [8] ,a taxi searching algorithm, and a scheduling algorithm. Supported by the index, the two algorithms quickly serve a large number of real time ride requests while reducing the travel distance of taxis compared with the case without taxi sharing.

This system considers and models monetary constraints in ridesharing. These constraints provide incentives not only for passengers but also for taxi drivers. passengers

ISSN: 2320-1363

will not pay more compared with no ridesharing and get compensated if their travel time is lengthened due to ridesharing. taxi drivers will make money for all the reroute distance due to ridesharing. The monetary constraints makes our modeling of the taxi ridesharing problem more realistic.

2. RELATED WORKS

The real time taxi sharing problem consists of a data model and constraints.

2.1 Data Model

2.1.1 Ride Request

A ride request Q is associated with a timestamp Q : t indicating when Q was submitted, a origin point Q:o, a destination point Q:d, a time window Q:pw defining the time interval. when the rider wants to be picked up at the origin point, and a time window Q:dw defining the time interval when the rider wants to be dropped off at the destination point. The early and late bounds of the pickup window are denoted by Q: pw: e and Q:pw:l respectively. Likewise, Q: dw:e and Q: dw:l stand for that of the delivery window. a rider only needs to explicitly indicate Q:d and Q: dw:l, as most information of a ride request can be automatically obtained from a rider's mobile phone, e.g.,Q:o and Q : t. we can assume that both Q:pw:e and Q: dw:e equals to Q : t, and Q:pw:l can be easily obtained by adding a fixed value, e.g., 5 minutes, to Q:pw:e.

2.1.2 Taxi Status

A taxi status V represents an instantaneous state of a taxi and is characterized by the following fields,

- V:ID. The unique identifier of the taxi.
- V:t. The time stamp associated with the status.
- V:l. The geographical location of the taxi at V:t.
- V .s. The current schedule of V , which is a temporally ordered sequence of origin and destination points of n ride requests Q1, Q2; . . .Qn such that for every ride request Qi, i ¼ 1; . . . ;n, either Qi : o precedes Qi : d in the sequence, or 2) only Qi:d exists in the sequence.
- V:r. The current projected route of V , which is a sequence of road network nodes calculated based on V:s.

From the definition, it is clear that the schedule of a vehicle status is dynamic, i.e., changes over time. For example, a schedule involving two ride requests Q1 and Q2 could be Q1:o → Q2:o → Q1:d → Q2 : d at a certain time. The schedule is updated to Q2:o → Q1:d → Q2 : d once the taxi has passed point Q1:o.

2.2 Constraints

The core of the taxi-sharing problem is to dispatch taxis to ride requests, subject to certain constraints. We say that a taxi status V satisfies a ride request Q or Q is satisfied by V if the following constraints are met.

2.2.1.Vehicle capacity constraint: The number of riders that sit in the taxi does not exceed the number of seats of a taxi at any time.

2.2.2.Time window constraints: All riders that are assigned to V should be able to depart from the origin point and arrive at the destination point during the corresponding pickup and delivery window, respectively.

2.2.3. Monetary constraints: These Constraints provide certain monetary incentives for both taxi

ISSN: 2320-1363

drivers and riders. That is, a rider does not pay more than without taxi sharing.a taxi driver does not earn less than without taxi sharing when travelling the same distance. The fare of existing riders decreases when a new rider joins the trip.

3 .SYSTEM ARCHITECTURE

The cloud consists of multiple servers for different purposes and a monitor for administers to oversee the running of the system. Taxi drivers and riders use the same smart phone App to interact with the system. Working steps of taxi ride sharing system.

1. A rider submits a new ride request Q to the Communication Server.All incoming ride requests of the system are streamed into a queue and then processed according to the first-come-first serve principle.

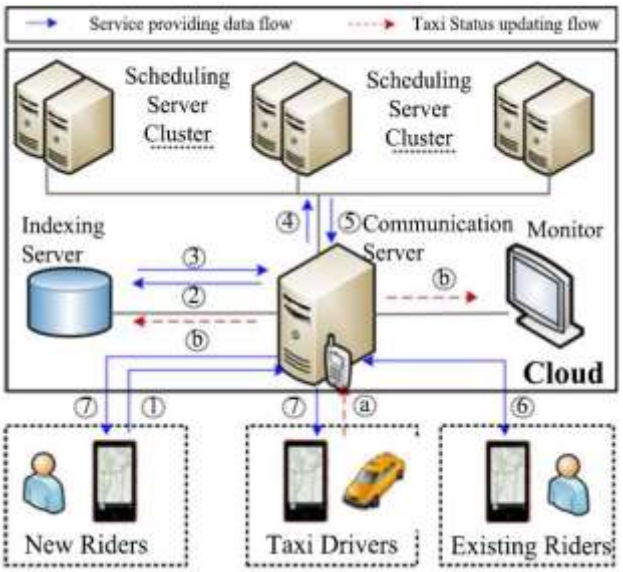

Fig .1. The architecture of the real time taxi ride sharing system.

2. For each ride request Q, the communication server sends it to the Indexing Server to search for candidate taxis SV that are likely to satisfy Q.
3. Using the maintained spatio-temporal index, the indexing server returns S_V to the communication server.
4. The communication server sends ride request Q and the received candidate taxi set S_V to the Scheduling Server Cluster.
5. The scheduling cluster checks whether each taxi in S_V can satisfy Q in parallel and returns the qualified taxi V that results in minimum increase in travel distance and a
detailed schedule
6. Each rider R who has been already assigned to the taxi V will be enquired whether they would like to accept the join of the new rider.
7. The information such as the estimated fare saving and travel time increase due to Q‘s join.Rider R accept or reject ride to driver.

4. TAXI SEARCHING

The taxi searching module quickly selects a small set of candidate taxis with the help of the spatio-temporal index.

The spatio-temporal index of taxis is built for speeding up the taxi searching process. Specifically, we partition the road network using a grid. each grid cell, we choose the road network node which is closest to the geographical centre of the grid cell as the anchor node of the cell[16]. The anchor node of a grid cell gi is thereafter denoted by ci. We compute the distance, denoted by d_{ij}, and travel time, denoted by t_{ij}, of the fastest path on the road network for each anchor node pair ci and cj. Both the distance and travel time is only computed once. (Alternatively, travel time can be updated dynamically, i.e. calculated once in a while (e.g., every 10 minutes) by leveraging

ISSN: 2320-1363

historical data archives, and travel time estimation techniques.

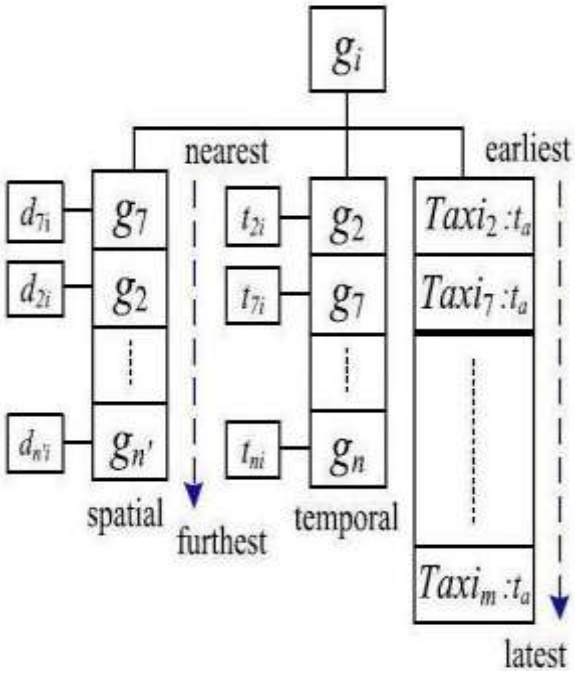

Fig .2. Spatio-temporal index of taxis.

The distance and travel time results are saved in a Matrix[16]. The matrix is there after referred to as the grid distance matrix. the grid distance matrix provides an approximation of the distance between any two nodes of the road network. These approximated distances avoid the expensive computation cost of frequent fastest path calculations at the stage of taxi searching.

When new GPS records are received from taxis, taxi lists need to be updated. Specifically, when a new GPS record from Vm is received, denote by gn the current cell in which Vm is located, the timestamp associated with Vm in the taxi list of cell gn and cells to be passed by Vm after gn need to be updated.

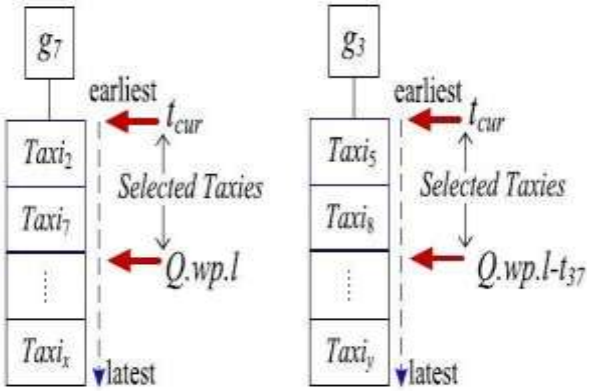

Fig .3. Choose taxis from the selected grid cells.

5. TAXI SCHEDULING

The set of taxi statuses SV retrieved for a ride request Q by the taxi searching algorithm, the purpose of the taxi scheduling process is to find the taxi status in SV which satisfies Q with minimum travel distance increase.To this end, given a taxi status, theoretically we need to try all possible ways of inserting Q into the schedule of the taxi status in order to choose the insertion which results in minimum increase in travel distance[16].

All possible ways of insertion can be created via three steps,

(i) reorder the points in the current schedule, subject to the precedence rule, i.e., any origin point precedes the corresponding destination point

(ii) insert Q:o into the schedule

(iii) insert the Q:d into the schedule.

The capacity and time window constraints are checked in all three steps, during which the insertion fails immediately if any constraint is violated. The monetary constraints are then checked for the insertion after

ISSN: 2320-1363

All three steps have been done successfully. Finally, among all insertions that satisfy all constraints, we choose the insertion that results in minimum increase in travel distance for the given taxi status.

Algorithm 1: computing the new schedule and route after an insertion

Data : ride request Q ,taxi status V, insertion position I for Q.o, insertion position j for Q.d ,current time t_{cur}

Result : Return new_ schedule if the insertion succeeds; otherwise return False.

if t_{cur} +(V.l → Q.o)>Q>pw.l **then** /*cannot arrive Q.o on time*/

return False

if the time delay incurred by the insertion of Q.o causes the slack time of any point after position i in schedule a smaller than 0 **then**

return False

new_ schedule ←Insert Q.o into V.s at position i /* the slack time of each pickup(delivery) point after position I is also updated accordingly */

t_j ← the scheduled arrival time of the j^{th} point of V.s

l_j ← the geographical location of the j^{th} point of V.s

if t_j +(l_j→Q.d)>Q.dw.l **then** /* cannot arrive Q.d on time */

return False

if the time delay incurred by the insertion of Q.d causes the slack time of any point after position j in new_schedule smaller than 0 **then** return False

New_schedule ← insert Q.d into new_schedule at position j /* the slack time of each pickup (delivery) point in new_schedule is also updated accordingly */

Return New_Schedule

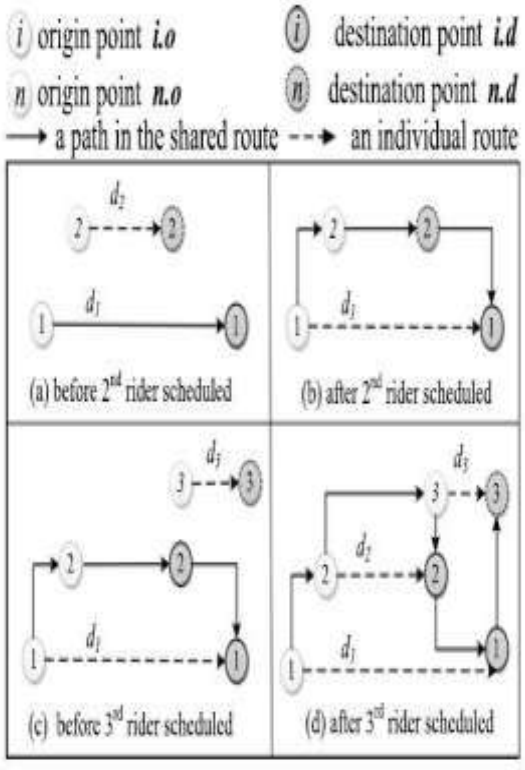

Fig.4. Example of ride sharing schedule.

6. REAL TIME TAXI RIDE SHARING FEATURES

The main objective of the taxi ridesharing application is to provide an easy to use interface for scheduling a pickup by the user and to view all the pickups that the user made using the application.

- Login (Passenger and Driver)
- Request the taxi for ride
- Showing the estimated duration, distance and fare to riders

ISSN: 2320-1363

- Showing the route map
- Add the new rider share rides with existing rider
- Send ride details to Email and SMS
- Schedule the rides
- Check taxi status
- Obtaining the current location using GPS
- Set a reminder notification
- Canceling a taxi
- View all ride request and pickup history

Advantages

- Google Cloud servers are fast and secured. Hence the information retrieval of the passenger is comparatively fast.
- Ride sharing feature is enabled on the proposed methodology in order to avoid the excess usage of the taxi.
- Our proposed system demonstrated its efficiency, effectiveness and scalability.

7. CONCLUSION

Cloud based real time taxi sharing methodology is proposed and implemented in this project. The data is synced with cloud servers which servers the data to mobile and web using the web services URL. The web service is a set of function which is deployed on the Google App Engine servers. The sharing methodology eliminates the cost of ride and it is proven to be time consuming where the users need not wait for the cab to reach the location from a new destination. the system saves the total travel distance of taxis when delivering passengers and the system can also save the taxi fare for each individual rider while the profit of taxi drivers does not decrease compared with the case where no taxi-sharing is conducted. The suggest that reordering the points of a schedule before the insertion of the new ride request for the purpose of travel distance minimization. Hence this mobile cloud based real time taxi ride sharing system enhances the new methodology for taxi riding. Cloud technology helps to retrieve the user information from any location. We presented detail interactions between end users (i.e. taxi riders and drivers) and the Cloud.

In the future, we consider incorporating the creditability of taxi drivers and riders into the taxi searching and scheduling algorithms. Additionally, we will further reduce the travel distance of taxis via ridesharing using advanced travel time estimation techniques .We also consider refining the ridesharing model by introducing social constraints, such as gender preference, habits preference (e.g., some people may prefer co-passengers who do not smoke).

REFERENCES

[1] Y. Dumas, J. Desrosiers, and F. Soumis, "The pickup and delivery problem with time windows," Eur. J. Oper. Res., vol. 54, no. 1, pp. 7–22, Sep. 1991.

[2] R. Baldacci, V. Maniezzo, and A. Mingozzi, "An exact method for the car pooling problem based on lagrangean column generation," Oper. Res., vol. 52, no. 3, pp. 422–439, 2004.

[3] R. W. Calvo, F. de Luigi, P. Haastrup, and V. Maniezzo, "A distributed geographic information system for the daily carpooling

ISSN: 2320-1363

problem," Comput. Oper. Res., vol. 31, pp. 2263–2278, 2004.

[4] K. Wong, I. Bell, and G. H. Michael, "Solution of the dial-a-ride problem with multi-dimensional capacity constraints," Int. Trans. Oper. Res., vol. 13, no. 3, pp. 195–208, May 2006.

[5] Z. Xiang, C. Chu, and H. Chen, "A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints," Eur. J. Oper. Res., vol. 174, no. 2, pp. 1117–1139, 2006.

[6] G. Gidofalvi and T. Pedersen, "Cab-sharing: An effective, door-to door,on-demand transportation service," in Proc. 6th Eur. Congr.Intell. Transp. Syst. Serv., 2007.

[7] K. Yamamoto, K. Uesugi, and T. Watanabe, "Adaptive routing of cruising taxis by mutual exchange of pathways," in Proc. 12th Int. Conf. Knowledge-Based Intell. Inf. Eng. Syst., Part II, 2008, pp. 559–566.

[8] D. Santani, R. K. Balan, and C. J. Woodard, "Spatio-temporal efficiency in a taxi dispatch system," Research Collection School Of Information Systems, Singapore Management University, Oct. 2008.

[9] E. Kamar and E. Horvitz, "Collaboration and shared plans in the open world: Studies of ridesharing," in Proc. 21st Int. Jont Conf. Artif. Intell., 2009, pp. 187–194.

[10] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: Driving directions based on taxi trajectories," in Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst., 2010, pp. 99–108.

[11] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G.-Z. Sun, "An interactive-voting based map matching algorithm," in Proc. 11th Int. Conf. Mobile Data Manage., 2010, pp. 43–52.

[11] P.-Y. Chen, J.-W. Liu, and W.-T. Chen, "A fuel-saving and pollution-reducing dynamic taxi-sharing protocol in VANETs," in Proc. IEEE 72nd Veh. Technol. Conf., Sep. 2010, pp. 1–5.

[12] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2011, pp. 316–324.

[13] W. Wu,W. S. Ng, S. Krishnaswamy, and A. Sinha, "To Taxi or Not to Taxi?—Enabling personalised and real-time transportation decisions for mobile users," in Proc. IEEE 13th Int. Conf. Mob. Data Manage., Jul. 2012, pp. 320–323.

[14] D. Zhang and T. He, "CallCab: A unified recommendation system for carpooling and regular taxicab services," in Proc. IEEE Int. Conf. Big Data, 2013, pp. 439–447.

[15] Y. Zheng, L. Capra, O. Wolfson, and H. Y.,"Urban Computing:Concepts, methodologies, and applications," ACM Trans. Intell.Syst. Technol., vol. 5, no. 3, article 38, Sep. 2014.

[16] Shuo Ma, Yu Zheng and Ouri Wolfson," Real-Time City-Scale Taxi Ridesharing," in Proc IEEE transactions on knowledge and data engineering., vol. 27, no. 7, pp. 1782-1795,July 2015.