iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** XI **Month of publication:** November 2017

DOI:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XI November 2017- Available at www.ijraset.com

A Novel APIs of Route-Saver Leveraging

M.Imadalli[1,] Munisekhar Prudhvi[2]
[1]M. Tech, Shree Engg. College, Tirupati
[2]M. Tech, Associate Professor, Shree Engg. College, Tirupati

Abstract: Mobile technology is one of the fast-growing technologies to improve the communication among the people. GPS-equipped vehicles can be regarded as mobile sensors probing traffic flows on road surfaces and users are usually experienced in finding the fastest (quickest) route to a destination based on query from the historical GPS trajectories of a large number of vehicles, and provide a user with the practically fastest route to a given destination at a given departure time. With the tremendous growth in mobile technology, people of the mobile user want to know the information regarding their needs which is based LBS (Location Based Service). LBS enable mobile users to query points-of-interest (e.g., ATM, restaurants) on various features (e.g., category, price, quality, and variety). They manage points-of-interest (POIs) specific to their applications, and enable mobile users to query for POIs that match with their preferences and time constraints and also it secures a user identity and locality within basic mobile communication services. For this approach this paper provides the survey about various techniques for providing the accurate and efficient query for the mobile users. This also provides the elapsed time to reach the destination. Service integration process helps to retrieve all service details from the server without inquiring query. In the proposed approach, this proposes a time-dependent service graph, where a node is a road segment frequently traversed by other users, to model the intelligence search of LBS and the properties of dynamic road networks. It is used to find the nearest neighbor services. We build the system based on a real world spatial simulation dataset and stored in MOD (moving object database).
Keywords: Time Dependent Service Graph, Spatial Data Mining, LBS, Elapsed Time, Intelligent Transportation Systems (ITS), Advanced Traffic Management Systems (ATMS), location based, services (LBS), Route-Saver module (RSM)

I. INTRODUCTION

Some existing works focus on retrieving individual objects by specifying a query consisting of a query location and a set of query keywords. These system define and categorize indoor distance between indoor uncertain objects and derive different distance bounds that can facilitate the join processing and better decision making which is also based on the shortest route. Indexing on road networks have been extensively studied in the existing. Various shortest path indices have been developed to support shortest path search only. After analyzing some existing system, all system only concentrated on the routes and traffic flow. But our proposed system gives better search like nearest services. The proposed system is used to find the nearest neighbor query and service. Based on the location and preference of the user the query answering can made. The proposed system develops a new rapid data access method called traffic based service selection. That is used to extend the road way in multidimensional method with the best algorithm. For example hotel details (food, price, room availability), theater (name of the films, ticket availability, etc.), etc.

II. RELATED WORK

we study the problem of KNN search on road networks. given a query location and a set of candidate objects in a road network, the KNN search finds the k nearest objects to the query location. to address this problem, we propose a balanced search tree index, called g-tree. the g-tree of a road network is constructed by recursively partitioning the road network into sub-networks and each g-tree node corresponds to a sub-network. inspired by classical KNN search on metric space, we introduce a best-first search algorithm on road networks, and propose an elaborately-designed assembly-based method to efficiently compute the minimum distance from a g-tree node to the query location. g-tree only take so (|v|log|v|) space, where |v| is the number of vertices in a network, and thus can easily scale up to large road networks with more than 20 millions vertices. Experimental results on eight real –world datasets show that our method significantly outperforms state-of-the-art methods, even by 2-3 orders of magnitude.

The k-nearest-neighbor (k-NN) query is one of the most popular spatial query types for location-based services (LBS). In this paper, we focus on k-NN queries in time-dependent road networks, where the travel time between two locations may vary significantly at different time of the day. In practice, it is costly for a LBS provider to collect real-time traffic data from vehicle sorroa dside sensors to compute the best route from a user to a spatial object of interest in terms of the travel time. Thus, we designs Mash Q, a serverside spatial mash up framework that enables a database server to efficiently evaluate K-NN queries using the route information and travel time accessed from an external Web mapping service, e.g., Microsoft Bing Maps. Due to the expensive cost and limitations of retrieving such external information, we propose three shared execution optimizations for Smash Q, namely, object grouping, direction sharing, and user grouping, to reduce the number of external Web mapping requests and provide highly accurate query answers. We evaluate SmashQ using Microsoft Bing Maps, a real road network, real data sets, and a synthetic data set. Experimental results show that SMashQ is efficient and capable of producing highly accurate query answers

A. Existing System

1) To meet the accuracy requirement, the frame work SMashQ is used for the LBS to answer KNN queries accurately by retrieving live travel times (and routes) from online route APIs (e.g., Google Directions API, Bing Maps API, which have live traffic information.
2) Indexing on road network shave been extensively studied in the literature. Various shortest path indices have been developed to support shortest path search efficiently.
3) Papa diaset al. study how to process range queries and KNN queries over points-of-interest, with respect to shortest path distances on a road network.
4) Thomsen et al. study the caching of shortest paths obtained from online route APIs. They exploit tthe optimal sub path property on cached paths to answer shortest path queries.

B. Disadvantages Of Existing System

1) Query results with inaccurate travel times may disrupt the users' schedules, cause their dissatisfaction, and eventually risk the LBS losing its users and advertisement revenues.
2) Similarly, high response time may drive users away from the LBS.
3) As a remark, online maps (e.g., Google Maps, Bing Maps), on the other hand, cannot process queries for the LBS either, because those queries may involve specific attributes (e.g., quality, price, facility) that are only maintained by the LBS.
4) SmashQ does not utilize route log to derive exact travel times nor lower/upper bounds to boost the query performance of the LBS.

C. Proposed System

1) In this proposed System, I exploit an observation namely that travel times change smoothly within a short duration. Routes recently obtained from online route APIs may still provide accurate travel times to answer current queries. This property enables us to design a more efficient solution for processing range and KNN queries.
2) Specifically, our method Route-Saver keeps at the LBS the routes which were obtained in the past d minutes (from an online route API), where d is the expiry time parameter. These recent routes are then utilized to derive lower/upper bounding travel times to reduce the number of route requests for answering range and KNN queries.
3) To reduce the number of route requests while providing accurate results, we combine information across multiple routes in the log to derive tight lower/upper bounding travel times. I also propose effective techniques to compute such bounds efficiently. Moreover, we examine the effect of different orderings for issuing route requests on saving route requests. And we study how to parallelize route requests in order to reduce the query response time further.

D. Advantages of Proposed System

1) Our experiments show that our solution is three times more efficient than SMashQ, and yet achieves high result accuracy (above 98 percent).
2) Combine information across multiple routes in the log to derive lower/upper bounding travel times, which support efficient and accurate range and KNN search.
3) Develop heuristics to parallelize route requests for reducing the query response time further.
4) Evaluate our solutions on a real route API and also on a simulated route API for scalability tests.

III. IMPLEMENTATION

A. Modules

1) Collect Hotel data

2) Route APIs
3) Travel times
4) Queries accurate

B. Module description

1) *Collect Hotel data:* Collecting time-series spatial data related to Hotel. The spatial data includes the hotel location, types of menus, cost and its ratings. Cost will be updated related to day to day life. All types of menus will be update. Based on the user opinions, ratings will be checked and updated. After get all details user can booking hotels.

C. Route APIs

An online route API has access to current traffic information . It takes a route request as input and then returns a route along with travel times on route segments. It provides mobile users with query services on a data set P, whose POIs (e.g., restaurants) are specific to the LBS's application. The LBS may store a road network G with edge weights as spatial distances, however G cannot provide live travel times. In case P and G do not fit in main memory, the LBS may store P as an R-tree and store the G as a disk-based adjacency list .The current travel time to p, it may also provide the current travel times to other objects p0 on the route, and may even offer tight ened lower/upper bounds of travel times to other objects.

D. Travel times

The time-tagged road network G and the route log L to derive lower and upper bounds of travel times for data points. As we will elaborate soon, these bounds enable us to save route requests during query processing .the live travel times of routes from online route APIs in order to offer accurate results. Maps to measure the live travel times for three pairs of locations .proposed for the LBS to answer KNN queries accurately by retrieving live travel times (and routes) from online route APIs.

E. Queries accurate

The user can acquire his current geo-location q and then issue queries to a location-based server. In this paper, we consider range and KNN queries based on live traffic. Upon receiving a KNN query from user q, the LBS first retrieves K objects with the small estnet work distance from q and issues route requests for them. Let g be the K smallest current travel time. User will get live traveling time and transport with perfect direction. User can see multiple hotels based on location. Update rating for all hotels and price cost related to menus.

IV. CONCLUSION AND FUTURE ENHANCEMENT

A. Conclusion

I propose a solution for the LBS to process range/KNN queries such that the query results have accurate travel times and the LBS incurs few number of route requests. Our solution Route-Saver collects recent routes obtained from an online route API (within d minutes). During query processing, it exploits those routes to derive effective lower-upper bounds for saving route requests, and examines the candidates for queries in an effective order. I have also studied the parallelization of route requests to further reduce query response time. Our experimental evaluation shows that Route-Saver is 3 times more efficient than a competitor, and yet achieves high result accuracy (above 98 percent).

B. Future Enhancement

I plan to investigate automatic tuning the expiry time d based on a given accuracy requirement. This would help the LBS guarantee its accuracy and improve their users' satisfaction.

V. REFERENCES

[1] 2011 Census TIGER/Line Shapefiles. (2011). [Online]. Available: http://www.census.gov/cgi-bin/geo/shapefiles2011/main
[2] 9th DIMACS Implementation Challenge on Shortest Paths. (2013). [Online]. Available: http://www.dis.uniroma1.it/challenge9/ data/tiger/
[3] Bing Data Suppliers. (2013). [Online]. Available: http://windows. microsoft.com/en-HK/windows-live/about-bing-data-supplie rs/
[4] Bing Maps API. (2013). [Online]. Available: http://www. microsoft.com/maps/developers/web.aspx
[5] Bing Maps Licensing and Pricing Information. (2013). [Online]. Available: http://www.microsoft.com/maps/product/licensing. aspx

[6] Google Directions & Bing Maps: Live Traffic Information. (2013). [Online]. Available: http://support.google.com/maps/bin/answer.py?hl=en&answer=2549020&topic=1687356&ctx=topic http://msdn.microsoft.com/en-us/library/aa907680.aspx

[7] Google Directions API. (2013). [Online]. Available: https:// developers.google.com/maps/documentation/directions/

[8] Google Directions API Usage Limits. (2013). [Online]. Available: https://developers.google.com/maps/faq#usagelimits

[9] Google Map Maker Data Download. (2013). [Online]. Available: https://services.google.com/fb/forms/mapmakerdatadownload/

[10] OpenStreetMap. (2013). [Online]. Available: http://www. openstreetmap.org/\

[11] Statistics of Usage. (2013). [Online]. Available: http://www. quantcast.com

[12] US Maps from Government. (2013). [Online]. Available: http:// www.usgs.gov/pubprod/

[13] N. Bruno, S. Chaudhuri, and L. Gravano, "STHoles: A multidimensional workload-aware histogram," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2001, pp. 211–222.

[14] E. P. F. Chan and Y. Yang, "Shortest path tree computation in dynamic graphs," IEEE Trans. Comput., vol. 58, no. 4, pp. 541–557, Apr. 2009.

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms. Cambridge, MA, USA: MIT Press, 2009.

[16] U. Demiryurek, F. B. Kashani, C. Shahabi, and A. Ranganathan, "Online computation of fastest path in time-dependent spatial networks," in Proc. 12th Int. Symp. Adv. Spatial Temporal Databases, 2011, pp. 92–111.

[17] A. Dingle and T. Partl, "Web cache coherence," Comput. Netw., vol. 28, pp. 907–920, 1996.

[18] M. Drozdowski, Scheduling for Parallel Processing, 1st ed. New York, NY, USA: Springer, 2009.

[19] H. Hu, D. L. Lee, and V. C. S. Lee, "Distance indexing on road networks," in Proc. 32nd Int. Conf. Very Large Data Bases, 2006, pp. 894–905.

[20] S. Jung and S. Pramanik, "An efficient path computation model for hierarchically structured topographical road maps," IEEE Trans. Knowl. Data Eng., vol. 14, no, 5, pp. 1029–1046, Sep./Oct. 2002.

[21] E. Kanoulas, Y. Du, T. Xia, and D. Zhang, "Finding fastest paths on a road network with speed patterns," in Proc. Int. Conf. Data Eng., 2006, p. 10.

[22] M. Kolahdouzan and C. Shahabi, "Voronoi-based K nearest neighbor search for spatial network databases," in Proc. 30th Int. Conf. Very Large Data Bases, 2004, pp. 840–851.

[23] H.-P. Kriegel, P. Kr€oger, P. Kunath, and M. Renz, "Generalizing the optimality of multi-step k -nearest neighbor query processing," in Proc. 10th Int. Symp. Adv. Spatial Temporal Databases, 2007, pp. 75–92.

[24] H.-P. Kriegel, P. Kr€oger, M. Renz, and T. Schmidt,, "Hierarchical graph embedding for efficient query processing in very large traffic networks," in Proc. 20th Int. Conf. Sci. Statis. Database Manage., 2008, pp. 150–167

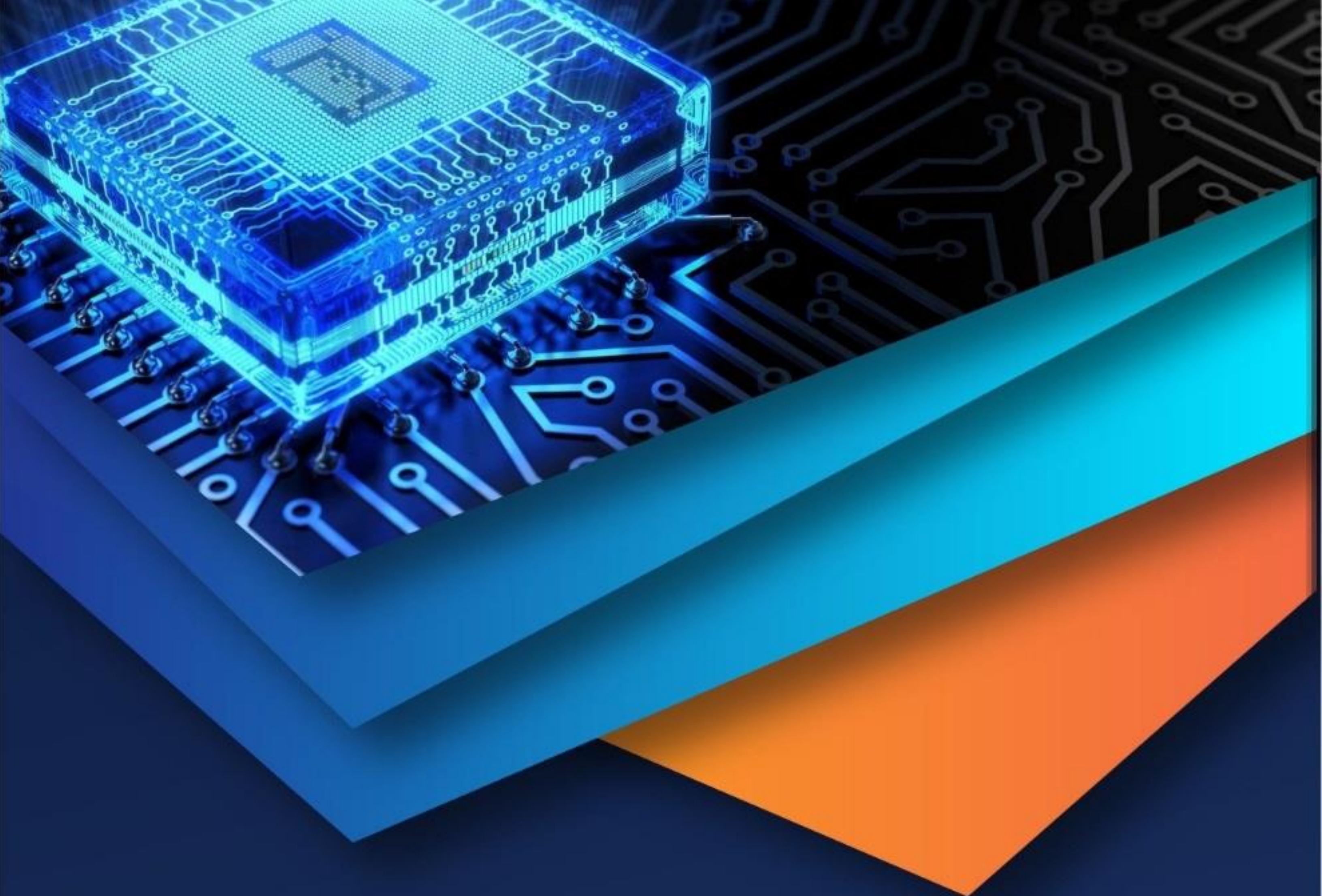
[25] H.-P. Kriegel, P. Kr€oger, M. Renz, and T. Schmidt, "Proximity queries in large traffic networks," in Proc. 15th Annu. ACM Int. Symp. Adv. Geographic Inform. Syst., 2007, p. 21.

[26] D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao, "Query processing in spatial network databases," in Proc. 29th Int. Conf. Very Large Data Bases, 2003, pp. 802–813.

[27] M. Qiao, H. Cheng, L. Chang, and J. X. Yu, "Approximate shortest distance computing: A query-dependent local landmark scheme," in Proc. IEEE 28th Int. Conf. Data Eng., 2012, pp. 462–473.

[28] H. Samet, J. Sankaranarayanan, and H. Alborzi, "Scalable network distance browsing in spatial databases," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2008, pp. 43–54.

[29] J. Sankaranarayanan and H. Samet, "Distance oracles for spatial networks," in Proc. IEEE Int. Conf. Data Eng., 2009, pp. 652–663.



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY