Veeresha et al. 2017. *Int. J. Vehicle Structures & Systems*, 9(2), 83-86
ISSN: 0975-3060 (Print), 0975-3540 (Online)
doi: 10.4273/ijvss.9.2.04

International Journal of Vehicle Structures & Systems
Available online at www.maftree.org/eja

A Novel Air Index for Range Queries in Road Networks

M. Veeresha[a] and M. Sugumaran[b]

Dept. of Computer Sci. and Engg., Pondicherry Engg. College, Pondicherry, India
[a]Corresponding Author, Email: veeresha.m@pec.edu,
[b]Email: sugu@pec.edu

ABSTRACT:

Objective of the present work is to improve range query performance using Hybrid Spatial Air Index (HSAI). HSAI has been designed with combination of both cache management and network coding for processing range queries in road networks. HSAI has been utilized the advantage of both cache management and network coding and reduce client search space. The experiments have been conducted for evaluating performance, the experimental results show that HSAI outperform.

KEYWORDS:

Hybrid spatial air index; Cache management; Network coding; Range queries; Road networks

CITATION:

M. Veeresha and M. Sugumaran. 2017. A Novel Air Index for Range Queries in Road Networks, *Int. J. Vehicle Structures & Systems,* 9(2), 83-86. doi:10.4273/ijvss.9.2.04.

1. Introduction

Query processing has been raised in many real-time and day-to-day applications due to advanced technology in wireless mobile environments. For example, a mobile user request nearest information through queries while moving in the road networks. The road networks have been modelled into time-independent and time-dependent, and computed travelling time based on the distance and traffic respectively. However, these models have been suffered from scalability and security due to point-to-point access strategy [1-5]. For improving scalability and security, wireless broadcast strategy has been adapted [6]. Wireless broadcast strategies are classified into push-based, pull-based and hybrid scheduling [7-8]. However, these strategies have been suffered from sequential data access. In Euclidian space, various spatial queries have been adapted [9-10]. An air index has been adapted for shortest path queries in road networks, but it doesn't address range query and k-nearest neighbour query [11]. NPI based air index adapted for various queries in road networks [12], but it doesn't address reliability, cache management.

To address these issues HSAI has been proposed for queries in road networks. HSAI has been utilized usages of network coding, cache management. The idea of this work has been explained as follows. The original road networks may contain number of data objects such as restaurants, shopping-malls, hospitals, gas-stations and schools. Based on client's requirements, data objects are classified into general and specific data sets, and compute cut-off point [13]. Using grid partition strategy, original road network is partitioned into small grid cells, and pre-computed the diameter of each cell and maximum/minimum network distance between every pair of cells that will be carried by HSAI. On the server site, we incorporate XOR-based network coding into broadcast scheduling program, and broadcast encoded general and specific data objects of HSAI with network connectivity information of each cell using Hybrid Broadcast (HB) scheduling. On the client site, we adapted Adaptive Cooperative Caching (ACC). In ACC, once a client receives a query request from a mobile user then it searches query result in its cache.

If not found then it sends request query to region Query Directory (QD). If region QD doesn't contain query result then it sends request query to other region QDs in the network. If none of the region QD do not respond then the client tune into channel, decode and retrieve required data and process the query using Dijkstra's shortest path algorithm. If required data doesn't broadcast while tuning then the client send request query to server via pull-based scheduling. If any free time-slot exists while broadcasting, then server broadcast encoded specific data objects (i.e., requested data objects) with general data objects of HSAI based on longest waiting time in the queue and priority order, and then updates optimal cut-off point. In this paper, HSAI and searching algorithm is proposed for range queries in road networks. Experiments are conducted to compare the performance of HSAI with state-of-the-art NPI.

2. Data dissemination via wireless channel using HSAI

Once data segment has been designed then server broadcast HSAI with data segment via wireless channel. If the client receives query request from the mobile user then it searches query result using ACC. The ACC strategy considers both spatial and temporal property of data objects when making cache replacement decisions based on client moment predictions. If query results not found in ACC then client tune into channel, retrieve required data and decode. Finally process the query using Dijkstr's shortest path algorithm. In wireless mobile environments, cache management and network coding has been improves query performance. In cache management, various cache replacement strategies have been considered either spatial or temporal property of data objects when making cache replacement decisions [14-18]. However, in order to improve query performance we have to consider both spatial and temporal property of data objects when making cache replacement decisions. In network coding, various coding strategies have been adapted for improving scalability and throughput [19-22]. However, in this work we propose XOR-based network coding because it is simple and efficient strategy compared to other network coding strategies.

3. Processing of range query using HSAI

This section describes searching algorithm of range query at client. The objective of range query algorithm is process the query efficiently and returns result to the client is shown in Fig. 1. For processing queries in road networks, Dijkstra's shortest path algorithm has been adapted because it is simple and efficient for small sub-graph retrieved by clients. In this work, assume that the clients' locations are located at network nodes and easily extend to support the case where the clients' locations are locating along the network edges.

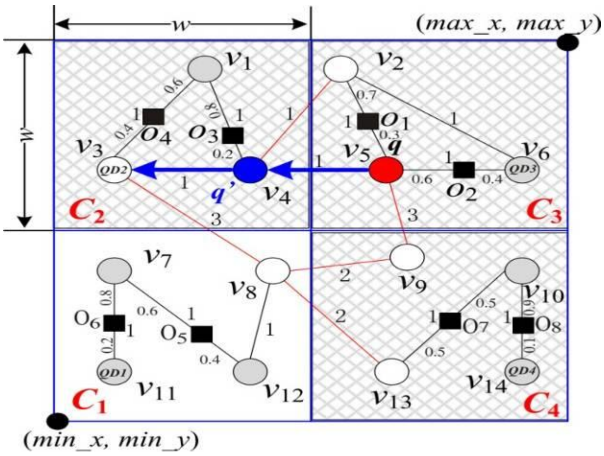

Fig. 1: Processing of range query using ACC and network coding

3.1. Range query

In road networks, range query retrieves all data objects within a network distance d from a query point q, and it denoted as $(q, d, S) = \{o \mid o \in S \wedge \|o, q\| \leq d\}$, where q is a query point, o is a data object, d is a network distance and S is a dataset in a road networks. The algorithm is presented as follows:

Procedure Range_query(q, d, S);
Input: *q – query point, d – distance, S – data set*
Output: *valid data objects within the range d*
▷*% receive_query( ) – a client waits for receiving a query from mobile user, t – time, C_q– valid grid cells from a query point q, R_q– valid range query result from a query point q, $\alpha_{q,i}$ – minimum or maximum network distance from a query point q %*
▷*% perform encoding and decoding at server and client sites respectively using XOR-network coding %*

```
begin
data set S divided into general and specific data sets;
compute a cut-off point cp;
query = receive_query( );
if query results found at client/region QD cache then
   return result;
else
   for each other region QD do
   if query result found at any region QD's cache then
         return result;
      else
         Listen_channel( );
      end if
   end for
end if
end
```

```
Procedure Listen_channel( );
begin
wait t seconds for required data;
 if a client required data doesn't broadcast then
      client send query to server;
      for each available time slot do
         if empty slot exist then
          select specific data objects based on waiting time & priority;
         end if
      end for
      update cut_off point cp;
else if required data is received from the server then
      HSAIHeader = retriveIndexHeader( );
      find out the grid cells C_q containing query q;
      read the q^th row R_q corresponds to C_q in matrix;
      for each cell C_i do
         if α_{q,i} <= d then add C_i to the candidate cell
          sort the candidate cells by their arrival times;
            for each candidate cell do
               SleepUntilCellBroadcast( );
               Listen_channel( );
               adjacencyLists = retriveCell( );
               subGraph.add(adjacencyLists);
               return Dijkstra(subGraph, q, d);
            end for
         end if
      end for
 end if
end
```

Assume that the client is located at node v_5 and request range query to find "four restaurants from query point q with network distance d = 3". Then the range query result is O_1, O_2, O_3 and O_4.

4 Performance evaluations

Experiments have been conducted for evaluating query performance using ns-2 simulator with window 7 platform, 2.33G Intel Core 2 CPU and 3.2 GB RAM.

4.1. Experimental setup

In this simulation, real road networks data set namely Oldenburg (OL) and California (CAL) have been considered. The OL contains 6,105 nodes and 7,035 edges, and CAL contains 21,048 nodes and 21,693 edges respectively [23]. The evolution is run on simulator which contains server, broadcast channel and clients. For simulating results, 100 clients and 500 random queries are used. In this work, we considered:

1) Size of data object is fixed at 128 bytes;
2) A set of data objects are randomly generated and uniformly distributed;
3) Query issuing points are always at the network nodes;
4) Network bandwidth is dynamic;
5) Tuning time and access latency are measured in terms of number of bytes of data transfer in a wireless channel.

4.2. Cycle length

In this simulation, cycle length plays an important because it is directly impacts on the access latency. The original road network is partitioning into $2^i \times 2^i$ uniform grid cells, where i vary from 0 to 4, (i.e., the number of grids ranges from 1, to 4, to 16, to 64, and to 256). The number of grid cells in a network is set to 4^i and then mapped into Hilbert curve order. The server disseminates data using (1, m) strategy by setting optimal m value [6]. In this work, the performance of HSAI compared with state-of-the-art NPI based on grid sizes such as 4^2, 4^3, and 4^4 respectively, and these are denoted as HSAI/NPI-16, HSAI/NPI-64 and HSAI/NPI-256. The parameter settings and broadcast cycle length are shown in Table 1 and Table 2 respectively.

Table 1: Parameter settings

Parameter	Values
k	1, 5, 10, 15
Query scope (d/D_N)	0.01, 0.05, 0.1, 0.2
Object density (|S|/|V|)	0.01, 0.05, 0.1, 0.2
No. of cells(N)	16, 64, 256

Table 2: Broadcast cycle length

Method	Index size (byte)	Data size (byte)	Cycle length (byte)
HSAI/NPI-16	2196	367764	389724
HSAI/NPI-64	33300	367764	700764
HSAI/NPI-256	526356	367764	5631324

4.3. Range query

In this experiment, an object density D_N is ignored but fixed at 0.1 because large number of data objects will not affects on query performance, and radius of range query is varied from $0.01D_N$, to $0.05D_N$, to $0.1D_N$ and to $0.2D_N$. By observing tuning time, with smaller radius, HSAI-16 and HSAI-64 have better tuning time compared to state-of-the-art NPI-16 and NPI-64 respectively. Even for range queries with large radius, the advantage of HSAI is still significant. By observing access latency, HSAI-16 has shorter access latency than state-of-the-art NPI-16 consistently. Similarly, HSAI-64 also smaller access latency than state-of-the-art NPI-64 in some cases because HSAI has been utilized network coding and cached data at clients. By observing experimental results of range queries as shown in Fig. 2, the performance of HSAI-64 has better than HSAI-16 because HSAI-16 has smaller index size hence shorter access latency. Similarly, HSAI-64 has narrow grid cells which provide tighter upper bound hence smaller search space, and resulting in shorter tuning time.

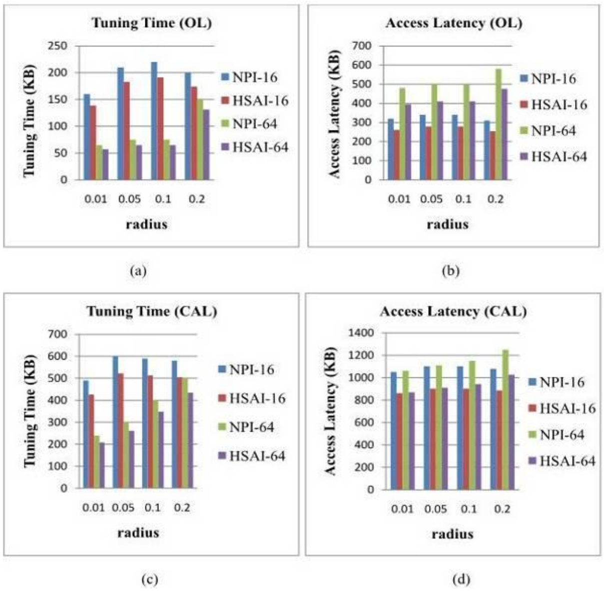

Fig. 2: Performance of range queries with radius using ACC and network coding

5. Conclusion

In this work, HSAI has been improved query performance compared to state-of-the-art NPI. HSAI has been reduced search space by using advantage of network coding and cache management. We can extent this work in future by adapting advanced network coding and cache management strategies.

REFERENCES:

[1] D. Zhang, C.Y. Chow, Q. Li, X. Zhang and Y. Xu. 2012. SMashQ: Spatial mashup framework for k-NN queries in time-dependent road networks, *Distributed Parallel Data bases*, 31, 259-287. https://doi.org/10.1007/s10619-012-7110-6.

[2] Y. Li, and M.L. Yiu. 2015. Route-saver: Leveraging route APIs for accurate and efficient query processing at location-based services, *IEEE Trans. Knowledge and Data Engg.*, 27, 235-249. https://doi.org/10.1109/TKDE.2014.2324597.

[3] H. Samet, J. Sankaranarayanan and H. Alborzi. 2008. Scalable network distance browsing in spatial databases, *Proc. ACM SIGMOD Int. Conf. on Mgmt Data*, 43-54. https://doi.org/10.1145/1376616.1376623.

[4] H. Kriegel, P. Kroger, P. Kunath, M. Renz and T. Schmidt. 2007. Proximity queries in large traffic networks, *Proc. 15th Annual ACM Int. Symp. Adv. Geographic Info. Systems*, 21-28. https://doi.org/10.1145/1341012.1341040.

[5] M.F. Mokbel, C.Y. Chow and W.G. Aref. 2006. The new casper: Query processing for location services without compromising privacy, *Proc. 32nd Int. Conf. Very Large Data Bases*, 763-774.

[6] T. Imielinski, S. Viswanathan and B.R. Badrinath. 1997. Data on air: Organization and access, *IEEE Trans. Knowledge and Data Engg.*, 9, 353-372. https://doi.org/10.1109/69.599926.

[7] G. Li, Q. Zhou and J. Li. 2015. A novel scheduling algorithm for supporting periodic queries in broadcast

environments, *IEEE Trans. Mobile Computing*, 14, 419-432. https://doi.org/10.1109/TMC.2015.2398417.

[8] W. Sun, Y. Qin, j. Wu, B. Zheng, Z. Zhang, P. Yu and J. Zhang. 2014. Air indexing for on-demand XML data broadcast, *IEEE Trans. Parallel and Distributed Systems*, 25, 1371-1381. https://doi.org/10.1109/TPDS.2013.87.

[9] K. Mouratidis, S. Bakiras and D. Papadias. 2009. Continuous monitoring of spatial queries in wireless broadcast environments, *IEEE Trans. Mobile Computing*, 8, 1297-1311. https://doi.org/10.1109/TMC.2009.14.

[10] B. Zheng, W.C. Lee and D.L. Lee. 2007. On searching continuous k-nearest neighbors in wireless data broadcast systems, *IEEE Trans. Mobile Computing*, 6, 748-761. https://doi.org/10.1109/TMC .2007.1004.

[11] U.L. Hou, H.J. Zhao, M.L. Yiu, Y. Li and Z. Gong. 2014. Towards online shortest path computation, *IEEE Trans. Knowledge and Data Engg.*, 26, 1012-1025. https://doi.org/10.1109/TKDE.2013.176.

[12] W. Sun, C. Chen, B. Zheng, C. Chen and P. Liu. 2015. An air index for spatial query processing in road networks, *IEEE Trans. Knowledge and Data Engg*., 27, 382-395. https://doi.org/10.1109/TKDE.2014.2330836.

[13] S. Kim and S.H. Kang. 2010. Scheduling data broadcast: An efficient cut-off point between periodic and on-demand data, *IEEE Communications Letters*, 14, 1176-1178. https://doi.org/10.1109/LCOMM.2010.101210.101228.

[14] P.T. Joy and K.P Jacob. 2012. A comparative study of cache replacement policies in wireless mobile networks, *Proc. Int. Symp. Adv. in Computing and Info. Tech.*, 609-619. https://doi.org/10.1007/978-3-642-31513-8_62.

[15] W.C. Peng and M.S. Chen. 2005. Design and performance studies of an adaptive cache retrieval scheme in a Mobile computing environment, *IEEE Trans. Mobile Computing*, 4, 29-40. https://doi.org/10.1109/TMC.2005.9.

[16] W.C. Peng and M.S. Chen. 2005. Shared data allocation in a mobile computing system: Exploring local and global optimization, *IEEE Trans. Parallel and Distributed Systems*, 16, 374-384. https://doi.org/10.1109/TPDS.2005.50.

[17] L. Yin and G. Cao. 2006. Supporting cooperative caching in adhoc networks, *IEEE Trans. Mobile Computing*, 5, 77-89. https://doi.org/10.1109/TMC.2006.15.

[18] Q. Zhu, D.L. Lee and W.C. Lee. 2011. Collaborative caching for spatial queries in mobile P2P networks, *IEEE Int. Conf. Data Engg.*, 279-290.

[19] R. Ahlswede, N. Cai, S.Y. Li and R.W. Yeung .2000. Network information flow, *IEEE Trans. Info. Theory*, 46, 1204-1216. https://doi.org/10.1109/18.850663.

[20] Y. Sagduyu and A. Ephremides. 2008. Cross-layer optimization of MAC and network coding in wireless queuing tandem networks, *IEEE Trans. Info. Theory*, 54, 554-571. https://doi.org/10.1109/TIT.2007.913423.

[21] Y. Birk and T. Kol. 2006. Coding on demand by an informed source (ISCOD) for efficient broadcast of different supplemental data to caching clients, *IEEE Trans. Info. Theory*, 52, 2825-2830. https://doi.org/10.1109/TIT.2006.874540.

[22] C. Zhan, C. Victor, S. Lee, J. Wang and Y. Xu. 2011. Coding-based data broadcast scheduling in on-demand broadcast, *IEEE Trans. Wireless Comms*., 10, 3774-3783. https://doi.org/10.1109/TWC.2011.092011.101652.

[23] F. Li, D. Cheng, M. Hadjieleftheriou, G. Kollios and S. Teng. 2005. On trip planning queries in spatial databases, *Proc. 9th Int. Conf. on Adv. Spatial Temporal Databases*, 923-923. https://doi.org/10.1007/11535331_16.