Query Processing Techniques for Big Spatial-Keyword Data*

Ahmed Mahmood
Purdue University
amahmoo@cs.purdue.edu

Walid G. Aref
Purdue University
aref@cs.purdue.edu

ABSTRACT

The widespread use of GPS-enabled cellular devices, i.e., smart phones, led to the popularity of numerous mobile applications, e.g., social networks, micro-blogs, mobile web search, and crowd-powered reviews. These applications generate large amounts of geo-tagged textual data, i.e., spatial-keyword data. This data needs to be processed and queried at an unprecedented scale. The management of spatial-keyword data at this scale goes beyond the capabilities of centralized systems. We live in the era of big data and the big data model is currently been used to address scalability issues in various application domains. This has led to the development of various *big spatial-keyword* processing systems. These systems are designed to ingest, store, index, and query huge amounts of spatial-keyword data. In this 1.5 hour tutorial, we explore recent research efforts in the area of big spatial-keyword processing. First, we give main motivations behind big spatial-keyword systems with real-life applications. We describe the main models for big spatial-keyword processing, and list the popular spatial-keyword queries. Then, we present the approaches that have been adopted in big spatial-keyword processing systems with special attention to data indexing and spatial and keyword data partitioning. Finally, we conclude this tutorial with a discussion on some of the open problems and research directions in the area of big spatial-keyword query processing.

CCS Concepts

•**Information systems** → **Data management systems; Parallel and distributed DBMSs; MapReduce-based systems; Geographic information systems;** *Query languages;*

*Walid G. Aref's research has been partially supported by the National Science Foundation under Grant III-1117766.

SIGMOD'17, May 14-19, 2017, Chicago, IL, USA

© 2017 ACM. ISBN 978-1-4503-4197-4/17/05. . . $15.00

DOI: http://dx.doi.org/10.1145/3035918.3054773

*** Part 1: Introduction and Background (10 minutes)**
- The scale of big spatial-keyword data
- Applications of big spatial-keyword data
- Limited support of big spatial-keyword data

***Part 2: Querying Big Spatial-Keyword Data (20 minutes)**
- Basic spatial-keyword queries
- Collective spatial-keyword queries
- Other queries
- Spatial-keyword query languages

***Part 3: Big Spatial-Keyword Processing Approaches (30 minutes)**
- Query specific algorithms
- Extensions to general-purpose big data systems
  - o Batch systems
  - o Stream processing systems
- Spatial-keyword only systems

***Part 4: Indexing in Big Spatial-Keyword Systems (10 minutes)**

***Part 5: Case Studies (10 minutes)**
- Batch systems case studies
- Streaming systems case studies
- Spatial-keyword services

***Part 6: Open Research (10 minutes)**
- Pipelined Evaluation of Spatial-keyword Queries
- Big Spatio-Temporal-keyword Processing
- Benchmarking Big Spatial-Keyword Systems
- Load-Balancing in Big Spatial-Keyword Systems

Figure 1: The outline of the tutorial (90 minutes).

Keywords

Spatial-keyword, Indexing, Systems, Big Data, Query Processing

1. INTRODUCTION

The unprecedented use of the GPS-enabled cellular devices has led to the popularity of numerous mobile applications. Examples of these applications include social networks, micro-blogs, mobile web-search, and crowd-powered reviews. Backed with a massive user base, these applications produce very large volumes of geo-tagged textual data, i.e., spatial-keyword data. For example, Facebook has on average 654 million active users. These users generate over 200 million daily messages [1, 40]. Also, there are about 800 million tweets that are being posted daily [3]. Several important applications depend on the processing of this spatial-keyword data, e.g., ad-targeting [56], micro-blogs analysis [53, 63], trip planning [12, 32, 50, 51], location-based recommendation [68, 83], and the discovery of new points of interest (POIs) [59].

Recently, several centralized systems and access methods have been introduced to process, index, and query spatial-keyword data, e.g., see [11, 12, 13, 14, 15, 16, 17, 19, 20, 24, 26, 27, 28, 33, 34, 35, 37, 38, 41, 42, 44, 45, 55, 61, 65, 69, 70, 72, 73, 75, 79, 84]. However, these centralized systems and access methods are not scalable to support the processing of big-data-scale amounts of spatial-keyword data. We live in the big data era, and recently several general-purpose big data systems, e.g., Hadoop [21], HBase [71], Spark [77], and Storm [67] have been developed for highly scalable data processing. However, these general-purpose big-data systems are not optimized to process spatial-keyword data. This has resulted in increased research effort to support scalable processing of big spatial-keyword data.

This 90-minute tutorial is focused on recent approaches adopted to support big spatial-keyword processing. We begin this tutorial with motivations for big spatial-keyword query processing. We describe the scale of data and list various applications that depend on the processing of spatial-keyword data. Subsequently, we formalize main spatial-keyword queries. Then, we survey existing approaches to process big spatial-keyword data in terms of the processing model, the queries supported, and the underlying big-data architecture. Finally, we highlight some of the open problems and research directions in the area of big spatial-keyword processing.

2. THE OUTLINE OF THE TUTORIAL

This **90-minute** tutorial is composed of six parts as illustrated in Figure 1. In the rest of this section, we detail the six parts of the tutorial.

2.1 Introduction and Background

In Part 1 of the tutorial, we spend 10 minutes introducing the problem domain of big spatial-keyword processing. We describe the massive scale of spatial-keyword data. We give example applications that require efficient processing of big spatial-keyword data. One important issue is that different applications demand different processing models. For example, *ad-targeting* requires *real-time* processing while the *discovery of POIs* can be performed in an *offline, i.e., batch* processing model. We show that existing general-purpose batch [21, 77] and stream [67, 78] processing systems are not optimized to support big spatial-keyword processing.

2.2 Querying Big Spatial-Keyword Data

Part 2 of the tutorial takes 20 minutes to formalize notations of spatial-keyword data and queries. Research in spatial-keyword processing has initially started in centralized environments, where numerous important spatial-keyword queries have been introduced and defined. Spatial-keyword queries can be classified [10, 15, 18, 57] into the following categories: *filter*, *top-k*, *collective*, and *other* queries.

- The *spatial-keyword filter* queries [15, 20, 31, 48, 56] identify data objects based on some spatial-keyword criteria, e.g., inside a specific spatial range and contain a specific set of keywords.
- The spatial-keyword *top-k* queries [20, 35, 38, 47, 62, 74] retrieve the most-relevant k objects. Ranking of objects is performed based on a function of the spatial distance and textual similarity between data objects and queries.
- The spatial-keyword *collective* queries [12, 28, 65, 80] identify groups of data object that collectively satisfy specific spatial and textual criteria, e.g., ones that collectively contain a set of keywords. A single data object by itself may not satisfy the group criteria.
- *Other* category includes other types of queries, e.g., spatial-keyword similarity distance and kNN join [9, 56], skyline [42], why-not [16], approximate [46], and reverse-kNN [47] queries.

A subset of these queries has been answered in the big spatial-keyword domain, e.g, spatial-keyword filter [5, 43, 50, 51, 52], top-k [36, 49, 68], collective [30, 32] and join [76, 82]. Moreover, in Part 2 of the tutorial, we also present the specifications of spatial-keyword query languages used to express queries over spatial-keyword data, e.g., see [2, 54, 57].

2.3 Big Spatial-Keyword Processing Approaches

Part 3 of this tutorial covers existing approaches to support big spatial-keyword processing that can be classified into the following categories:

- **Query specific algorithms** [30, 43, 76, 82] are developed to address certain query types. These algorithms run on general-purpose big-data systems without any changes to the underlying structure of the system or the layout of the data being processed. These algorithms typically group data objects for processing according to their spatial and/or textual properties.
- **Extensions to general-purpose big-data systems** are introduced to make big-data systems aware of the specific nature of spatial-keyword data. Typically, these extensions include indexing and partitioning techniques. These extensions are tailored to the architecture of the underlying big-data system. Extensions to main architectures of big-data systems can be classified as follows: (1) *Extensions to disk-based batch systems* [36, 52, 59], (2) *Extensions to Resilient Distributed Datasets (RDD) systems, e.g., Spark* [32], and (3) *Extensions to real-time streaming systems, e.g., Storm* [56, 68].
- **Spatial-keyword-only systems** [53, 63, 66] are designed from scratch and are fully optimized for the sole purpose of big spatial-keyword processing. These systems typically do not support other types of data or queries.

Most of these systems assume that the geo-location model, i.e., the locations of the data objects and the queries are defined using the longitude and latitude coordinates. However, other systems answer spatial-keyword queries assuming a graph model, i.e., a road network, e.g., see [30, 50, 51].

2.4 Indexing in Big Spatial-Keyword Systems

In Part 4 of the tutorial, we spend 10 minutes describing the main indexing and partitioning techniques that are adopted in big spatial-keyword systems. In batch processing systems [5, 36, 52], spatial-keyword data objects are indexed using a combination of a spatial-index, e.g., R-tree [29], Grid, K-D tree [8, 60] or Quad-tree [25], and a textual index, e.g., inverted lists [85], or ordered-keyword-tries [22]. These indexes are often realized on top of HDFS [64].

In a distributed streaming environment [56, 68], incoming spatial-keyword data objects are routed to specific worker processes. This routing can be based on the spatial or textual properties of the incoming data. In this part of the tutorial, we survey main indexing, and partitioning techniques of spatial-keyword data under both the batch and streaming system architectures.

2.5 Case Studies

In Part 5 of the tutorial, we spend 10 minutes presenting example big spatial-keyword processing systems. We give examples of both batch [50, 51, 52] and streaming [56, 68] big spatial-keyword systems. Our discussion of these systems focuses on the queries answered, the underlying architecture, the indexing and data partitioning techniques adopted in the aforementioned systems. We give examples of services that rely on general-purpose big-data systems for spatial-keyword processing. For example, Nimbus [40], a service for tuning predicates of spatial-keyword queries over tweets, uses Spark Streaming [78] to ingest and process incoming tweets. Also, sksOpen [49, 81], a service for querying and visualizing spatial-keyword data, uses MapReduce to reduce the time needed for spatial-keyword indexing. Yang et al., [76] propose to offload the processing of spatial-keyword join queries within wireless sensor networks (WSN) to a MapReduce cluster. ModisSENSE [59] uses MapReduce to run clustering algorithms on big spatial-keyword data to identify top-k POIs based on user feedback from tweets and social networks.

2.6 Future Research Directions

Big spatial-keyword processing is becoming a hot topic. In Part 6 of the tutorial, we spend 10 minutes highlighting some research directions in big spatial-keyword processing:

Pipelined Evaluation of Spatial-keyword Queries
Current spatial-keyword system proposals typically support small subsets of spatial-keyword queries (usually one or two). A typical system uses tailored query-specific algorithms and indexes. While many spatial-keyword queries, e.g., skyline, why-not, and preference queries have been addressed for a centralized environment, these queries have not been investigated for big-data processing platforms. Traditional RDBMSs have several building block-operators that are composable to form complex queries. Then, these queries are optimized and are evaluated in a pipelined manner. Pipelined evaluation for complex spatial-keyword queries in big-data systems remains an open research direction for both batch and streaming environments.

Big Spatio-Temporal-keyword Processing
Big spatial-only systems use some sort of distributed spatial indexing. Typically, spatial-keyword data involves a temporal dimension, e.g., the timestamp of a tweet or a web search. It is important to have a big spatio-temporal and keyword system that is able to process, index, and query spatial-temporal-keyword data in a scalable manner. Also, existing big spatial-keyword streaming systems do not support processing over a time-sliding window, e.g., continuous aggregation of data. This type of data management requires novel distributed spatial-temporal-keyword indexing, query processing algorithms, query languages, and visualization schemes that account for the peculiarities of the temporal dimension.

Benchmarking Big Spatial-Keyword Systems
Currently, many researchers use synthetic and domain-specific datasets, e.g., micro-blogs datasets, in the evaluations of their proposed big spatial-keyword systems and algorithms. Several benchmarks and data generators exist for spatial-only [58] and relational evaluation [4]. Existing spatial-keyword benchmarks are either small in size with limited queries [15] or focus on a specific use cases, e.g., social network analysis as proposed by Doudali et al. [23]. This calls for the development of a large-scale spatial-keyword benchmark that includes large datasets and various realistic queries for both batch and streaming environments. This benchmark is essential for the effective evaluation of emerging systems.

Load-Balancing in Big Spatial-Keyword Systems
Big-data systems distribute the processing across multiple processes. It is typical that the distribution of workload in spatial-keyword data be skewed and does not follow a uniform distribution. This calls for load-balancing techniques that ensure fair workload distribution of the worker processes. Existing load-balancing approaches in spatial-only big-data systems [6] are not directly applicable to big spatial-keyword systems. The only proposal for adaptivity in big a spatial-keyword system [56] assumes only a streaming environment under specific types of queries. This calls for a general load-balancing technique that is applicable in batch and streaming systems and is effective under various query workloads.

3. TARGETED AUDIENCE AND PREREQUISITES

This tutorial targets researchers and developers that are interested in conducting research in big spatial-keyword systems. This tutorial does not require prior knowledge about spatial-keyword data or big-data systems and only requires basic knowledge of the components of database systems, e.g., SQL and indexing.

4. RELEVANCE TO SIGMOD

Spatial-keyword processing and big-data processing are two important topics that are currently investigated by the database research community. Many research related to spatial-keyword processing [5, 12, 28, 37, 47, 59] and big data systems [7, 39, 67] are published in SIGMOD. Since this tutorial discusses big spatial-keyword processing, the tutorial is relevant to the database and data management community and to the SIGMOD conference.

5. PRESENTERS BIOGRAPHY

Ahmed Mahmood is a Ph.D. candidate at the Department of Computer Science, Purdue University. His research interests are spatial-keyword and distributed stream processing. He has been awarded the Purdue CS Teaching Fellowship, the Teaching Academy Graduate Teaching Award, and the Raymond Boyce Graduate Teacher Award. Ahmed is the main designer and developer of Tornado; the first distributed spatial-keyword stream processing system. For more information, please visit: http://www.cs.purdue.edu/homes/amahmoo

Walid G. Aref is a professor of computer science at Purdue. His research interests are in the areas of spatial and spatio-temporal data systems, data streaming, indexing, and query processing techniques. His research has been supported by the NSF, the National Institute of Health, Purdue Research Foundation, Qatar National Research Foundation, CERIAS, Panasonic, and Microsoft Corp. In 2001, he received the CAREER Award from the National Science Foundation and in 2004, he received a Purdue University Faculty Scholar award. He has received several best-paper awards including the 2016 VLDB 10-Year Best-Paper award. Walid is an associate editor of the ACM Transactions of Database Systems (ACM TODS) and the ACM Transactions of Spatial Algorithms and Systems (TSAS), and has been an editor of the VLDB Journal. He is an executive committee member, co-founder, and the past chair of the ACM SIGSPATIAL. For more information, please visit: https://www.cs.purdue.edu/people/aref.

References

[1] Facebook states. http://newsroom.fb.com/company-info/.

[2] GNIP. http://support.gnip.com/apis/powertrack/overview.html.

[3] Internet live stats. https://internetlivestats.com/.

[4] The TPC-H benchmark. http://www.tpc.org/tpch/.

[5] L. Alarabi, A. Eldawy, R. Alghamdi, and M. F. Mokbel. Tareeg: a mapreduce-based web service for extracting spatial data from openstreetmap. In *SIGMOD*, pages 897–900. ACM, 2014.

[6] A. M. Aly, A. R. Mahmood, M. S. Hassan, W. G. Aref, M. Ouzzani, H. Elmeleegy, and T. Qadah. Aqwa: adaptive query workload aware partitioning of big spatial data. *PVLDB*, 8(13):2062–2073, 2015.

[7] M. Armbrust, R. S. Xin, C. Lian, Y. Huai, D. Liu, J. K. Bradley, X. Meng, T. Kaftan, M. J. Franklin, A. Ghodsi, et al. Spark sql: Relational data processing in spark. In *SIGMOD*, pages 1383–1394. ACM, 2015.

[8] M. J. Berger and S. H. Bokhari. A partitioning strategy for nonuniform problems on multiprocessors. *IEEE Transactions on Computers*, 100(5):570–580, 1987.

[9] P. Bouros, S. Ge, and N. Mamoulis. Spatio-textual similarity joins. *VLDB*, 6(1):1–12, 2012.

[10] X. Cao, L. Chen, G. Cong, C. S. Jensen, Q. Qu, A. Skovsgaard, D. Wu, and M. L. Yiu. Spatial keyword querying. In *Conceptual Modeling*, pages 16–29. 2012.

[11] X. Cao, G. Cong, T. Guo, C. S. Jensen, and B. C. Ooi. Efficient processing of spatial group keyword queries. *TODS*, 40(2):13, 2015.

[12] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi. Collective spatial keyword querying. In *SIGMOD*, pages 373–384, 2011.

[13] A. Cary, O. Wolfson, and N. Rishe. Efficient and scalable method for processing top-k spatial boolean queries. In *Scientific and Statistical Database Management*, pages 87–95, 2010.

[14] L. Chen, G. Cong, X. Cao, and K.-L. Tan. Temporal spatial-keyword top-k publish/subscribe. In *ICDE*, pages 255–266, 2015.

[15] L. Chen, G. Cong, C. S. Jensen, and D. Wu. Spatial keyword query processing: An experimental evaluation. In *PVLDB*, volume 6, pages 217–228, 2013.

[16] L. Chen, X. Lin, H. Hu, C. S. Jensen, and J. Xu. Answering why-not questions on spatial keyword top-k queries. In *ICDE*, pages 279–290, 2015.

[17] M. Christoforaki, J. He, C. Dimopoulos, A. Markowetz, and T. Suel. Text vs. space: efficient geo-search query processing. In *CIKM*, pages 423–432, 2011.

[18] G. Cong and C. S. Jensen. Querying geo-textual data: Spatial keyword queries and beyond. In *SIGMOD*, pages 2207–2212. ACM, 2016.

[19] G. Cong, C. S. Jensen, and D. Wu. Efficient retrieval of the top-k most relevant spatial web objects. *VLDB*, 2(1):337–348, 2009.

[20] I. De Felipe, V. Hristidis, and N. Rishe. Keyword search on spatial databases. In *ICDE*, pages 656–665, 2008.

[21] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[22] E. K. Donald. The art of computer programming. *Sorting and searching*, 3:426–458, 1999.

[23] T.-D. DOUDALI. Performance evaluation of social networking services using a spatio-temporal and textual big data generator.

[24] J. Fan, G. Li, L. Zhou, S. Chen, and J. Hu. Seal: Spatio-textual similarity search. *PVLDB*, 5(9):824–835, 2012.

[25] R. A. Finkel and J. L. Bentley. Quad trees a data structure for retrieval on composite keys. *Acta informatica*, 4(1):1–9, 1974.

[26] Y. Gao, Y. Wang, and S. Yi. Preference-aware top-k spatio-textual queries. In *WAIM*, pages 186–197. Springer, 2016.

[27] R. Göbel, A. Henrich, R. Niemann, and D. Blank. A hybrid index structure for geo-textual searches. In *CIKM*, pages 1625–1628. ACM, 2009.

[28] T. Guo, X. Cao, and G. Cong. Efficient algorithms for answering the m-closest keywords query. In *SIGMOD*, pages 405–418, 2015.

[29] A. Guttman. *R-trees: a dynamic index structure for spatial searching*, volume 14. ACM, 1984.

[30] Y. Hao, H. Cao, Y. Qi, C. Hu, S. Brahma, and J. Han. Efficient keyword search on graphs using mapreduce. In *Big Data*, pages 2871–2873. IEEE, 2015.

[31] R. Hariharan, B. Hore, C. Li, and S. Mehrotra. Processing spatial-keyword (sk) queries in geographic information retrieval (gir) systems. In *SSBDM*, pages 16–16, 2007.

[32] P. He, H. Xu, X. Zhao, and Z. Shen. Scalable collective spatial keyword query. In *ICDEW*, pages 182–189. IEEE, 2015.

[33] T.-A. Hoang-Vu, H. T. Vo, and J. Freire. A unified index for spatio-temporal keyword queries. In *CIKM*, pages 135–144. ACM, 2016.

[34] H. Hu, G. Li, Z. Bao, J. Feng, Y. Wu, Z. Gong, and Y. Xu. Top-k spatio-textual similarity join. *TKDE*, 28(2):551–565, 2016.

[35] W. Huang, G. Li, K.-L. Tan, and J. Feng. Efficient safe-region construction for moving top-k spatial keyword queries. In *CIKM*, pages 932–941, 2012.

[36] J. Jiang, H. Lu, B. Yang, and B. Cui. Finding top-k local users in geo-tagged social media data. In *ICDE*, pages 267–278. IEEE, 2015.

[37] M. Jiang, A. W.-C. Fu, and R. C.-W. Wong. Exact top-k nearest keyword search in large networks. In *SIGMOD*, pages 393–404. ACM, 2015.

[38] A. Khodaei, C. Shahabi, and C. Li. Hybrid indexing and seamless ranking of spatial and textual features of web documents. In *DEXA*, pages 450–466, 2010.

[39] S. Kulkarni, N. Bhagat, M. Fu, V. Kedigehalli, C. Kellogg, S. Mittal, J. M. Patel, K. Ramasamy, and S. Taneja. Twitter heron: Stream processing at scale. In *SIGMOD*, pages 239–250. ACM, 2015.

[40] C.-A. Lai, J. Donahue, A. Musaev, and C. Pu. Nimbus: tuning filters service on tweet streams. In *2015 IEEE International Congress on Big Data*, pages 623–630. IEEE, 2015.

[41] G. Li, J. Feng, and J. Xu. Desks: Direction-aware spatial keyword search. In *ICDE*, pages 474–485, 2012.

[42] J. Li, H. Wang, J. Li, and H. Gao. Skyline for geo-textual data. *GeoInformatica*, 20(3):453–469, 2016.

[43] W. Li, W. Wang, and T. Jin. Evaluating spatial keyword queries under the mapreduce framework. In *DASFAA*, pages 251–261. Springer, 2012.

[44] Z. Li, K. C. Lee, B. Zheng, W.-C. Lee, D. Lee, and X. Wang. Ir-tree: An efficient index for geographic document search. *TKDE*, 23(4):585–599, 2011.

[45] S. Liu, G. Li, and J. Feng. Star-join: spatio-textual similarity join. In *CIKM*, pages 2194–2198. ACM, 2012.

[46] Z. Liu. Spatial approximate keyword query processing in cloud computing system. *International Journal of Database Theory and Application*, 8(2):81–94, 2015.

[47] J. Lu, Y. Lu, and G. Cong. Reverse spatial and textual k nearest neighbor search. In *SIGMOD*, pages 349–360, 2011.

[48] Y. Lu, J. Lu, G. Cong, W. Wu, and C. Shahabi. Efficient algorithms and cost models for reverse spatial-keyword k-nearest neighbor search. *TODS*, 39(2):13, 2014.

[49] Y. Lu, M. Zhang, S. Witherspoon, Y. Yesha, Y. Yesha, and N. Rishe. sksopen: efficient indexing, querying, and visualization of geo-spatial big data. In *ICMLA*, volume 2, pages 495–500. IEEE, 2013.

[50] S. Luo, Y. Luo, S. Zhou, G. Cong, and J. Guan. Disks: a system for distributed spatial group keyword search on road networks. *VLDB*, 5(12):1966–1969, 2012.

[51] S. Luo, Y. Luo, S. Zhou, G. Cong, J. Guan, and Z. Yong. Distributed spatial keyword querying on road networks. In *EDBT*, pages 235–246. Citeseer, 2014.

[52] Y. Ma, Y. Zhang, and X. Meng. St-hbase: a scalable data management system for massive geo-tagged objects. In *WAIM*, pages 155–166. Springer, 2013.

[53] A. Magdy, L. Alarabi, S. Al-Harthi, M. Musleh, T. M. Ghanem, S. Ghani, and M. F. Mokbel. Taghreed: a system for querying, analyzing, and visualizing geotagged microblogs. In *SIGSPATIAL*, pages 163–172, 2014.

[54] A. Magdy and M. F. Mokbel. Towards a microblogs data management system. In *MDM*, volume 1, pages 271–278, 2015.

[55] A. Magdy, M. F. Mokbel, S. Elnikety, S. Nath, and Y. He. Venus: Scalable real-time spatial queries on microblogs with adaptive load shedding. *TKDE*, 28(2):356–370, 2016.

[56] A. R. Mahmood, A. M. Aly, T. Qadah, E. K. Rezig, A. Daghistani, A. Madkour, A. S. Abdelhamid, M. S. Hassan, W. G. Aref, and S. Basalamah. Tornado: A distributed spatio-textual stream processing system. *PVLDB*, 8(12):2020–2023, 2015.

[57] A. R. Mahmood, W. G. Aref, A. M. Aly, and M. Tang. Atlas: On the expression of spatial-keyword group queries using extended relational constructs. In *SIGSPATIAL*, pages 45:1–45:10, 2016.

[58] M. F. Mokbel, L. Alarabi, J. Bao, A. Eldawy, A. Magdy, M. Sarwat, E. Waytas, and S. Yackel. MNTG: an extensible web-based traffic generator. 2013.

[59] I. Mytilinis, I. Giannakopoulos, I. Konstantinou, K. Doka, D. Tsitsigkos, M. Terrovitis, L. Giampouras, and N. Koziris. Modissense: A distributed spatio-temporal and textual processing platform for social networking services. In *SIGMOD*, pages 895–900. ACM, 2015.

[60] B. C. Ooi, K. J. McDonell, and R. Sacks-Davis. Spatial kd-tree: An indexing mechanism for spatial databases. In *IEEE COMPSAC*, volume 87, page 85, 1987.

[61] J. B. Rocha-Junior, O. Gkorgkas, S. Jonassen, and K. Nørvåg. Efficient processing of top-k spatial keyword queries. In *International Symposium on Spatial and Temporal Databases*, pages 205–222. Springer, 2011.

[62] J. B. Rocha-Junior and K. Nørvåg. Top-k spatial keyword queries on road networks. In *EDBT*, pages 168–179, 2012.

[63] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling. Twitterstand: news in tweets. In *SIGSPATIAL*, pages 42–51. ACM, 2009.

[64] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. The hadoop distributed file system. In *MSST*, pages 1–10. IEEE, 2010.

[65] A. Skovsgaard and C. S. Jensen. Finding top-k relevant groups of spatial web objects. *The VLDB Journal*, 24(4):537–555, 2015.

[66] B. E. Teitler, M. D. Lieberman, D. Panozzo, J. Sankaranarayanan, H. Samet, and J. Sperling. Newsstand: a new view on news. In *SIGSPATIAL*, page 18. ACM, 2008.

[67] A. Toshniwal, S. Taneja, A. Shukla, K. Ramasamy, J. M. Patel, S. Kulkarni, J. Jackson, K. Gade, M. Fu, J. Donham, et al. Storm@ twitter. In *SIGMOD*, pages 147–156. ACM, 2014.

[68] X. Wang, W. Zhang, Y. Zhang, X. Lin, and Z. Huang. Top-k spatial-keyword publish/subscribe over sliding window. *The VLDB Journal*, pages 1–26, 2016.

[69] X. Wang, Y. Zhang, W. Zhang, X. Lin, and Z. Huang. Skype: top-k spatial-keyword publish/subscribe over sliding window. *PVLDB*, 9(7):588–599, 2016.

[70] X. Wang, Y. Zhang, W. Zhang, X. Lin, and W. Wang. Ap-tree: Efficiently support continuous spatial-keyword queries over stream. In *ICDE*, pages 1107–1118, 2015.

[71] H. Wiki. Hbase: bigtable-like structured storage for hadoop hdfs.

[72] D. Wu and C. S. Jensen. A density-based approach to the retrieval of top-k spatial textual clusters. In *CIKM*, pages 2095–2100. ACM, 2016.

[73] D. Wu, M. L. Yiu, G. Cong, and C. S. Jensen. Joint top-k spatial keyword query processing. *TKDE*, 24(10):1889–1903, 2012.

[74] D. Wu, M. L. Yiu, C. S. Jensen, and G. Cong. Efficient continuously moving top-k spatial keyword query processing. In *ICDE*, pages 541–552, 2011.

[75] J. Yang, W. Zhang, Y. Zhang, X. Wang, and X. Lin. Categorical top-k spatial influence query. *WWW*, pages 1–29, 2016.

[76] M. Yang, L. Zheng, Y. Lu, M. Guo, and J. Li. Cloud-assisted spatio-textual k nearest neighbor joins in sensor networks. In *INISCom*, pages 12–17. IEEE, 2015.

[77] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *USENIX*, pages 2–2. USENIX Association, 2012.

[78] M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, and I. Stoica. Discretized streams: Fault-tolerant streaming computation at scale. In *ACM Symposium on Operating Systems Principles*, pages 423–438. ACM, 2013.

[79] C. Zhang, Y. Zhang, W. Zhang, and X. Lin. Inverted linear quadtree: Efficient top k spatial keyword search. *TKDE*, 28(7):1706–1721, 2016.

[80] D. Zhang, Y. M. Chee, A. Mondal, A. K. Tung, and M. Kitsuregawa. Keyword search in spatial databases: Towards searching by document. In *ICDE*, pages 688–699, 2009.

[81] M. Zhang, H. Wang, Y. Lu, T. Li, Y. Guang, C. Liu, E. Edrosa, H. Li, and N. Rishe. Terrafly geocloud: an online spatial data analysis and visualization system. *TIST*, 6(3):34, 2015.

[82] Y. Zhang, Y. Ma, and X. Meng. Efficient spatio-textual similarity join using mapreduce. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 1, pages 52–59. IEEE, 2014.

[83] Y. Zhang, Y.-Z. Ma, and X.-F. Meng. Efficient processing of spatial keyword queries on hbase. *Journal of Chinese Computer Systems*, 33(10):2141–2146, 2012.

[84] K. Zheng, B. Zheng, J. Xu, G. Liu, A. Liu, and Z. Li. Popularity-aware spatial keyword search on activity trajectories. *World Wide Web*, pages 1–25, 2016.

[85] J. Zobel and A. Moffat. Inverted files for text search engines. *ACM computing surveys (CSUR)*, 38(2):6, 2006.