Spatial Data Systems Support for the Internet of Things - Challenges and Opportunities

Mohamed Sarwat
Arizona State University
Tempe, USA
msarwat@asu.edu

Abstract

The Internet of Things (IoT) has recently received significant attention. An IoT device may possess an array of sensors that for example monitors the air temperature, carbon monoxide level, wifi signals, and sound intensity. IoT data is initially created on the device, then sent over to a central database system (e.g., the cloud) that organizes and prepares such data for the ongoing use by myriad applications, which include but are not limited to smart home, smart city, the industrial internet, connected cars, and connected health. Data generated by IoT devices is inherently spatial and temporal. For instance, an audio signal represents the variation of the sound intensity (retrieved by a sound sensor) over the time dimension. Furthermore, IoT devices are either installed in a static location (e.g., a building, a traffic intersection) or can be attached to moving objects such as a connected vehicle or a wearable device. In this article, we argue that existing IoT data systems do not properly consider the SpatioTemporal aspect of such data. Hence, the article represents a call for action to the SIGSPATIAL community in order to conduct research on building systems and applications that treat both the spatial and temporal dimensions of IoT data as first class citizens.

1 Motivation

The Internet of Things (IoT) has recently received significant attention from both industry and academia. The Boston Consulting Group predicts that by 2020, $267 Billion will be spent on IoT technologies, products, and services [1]. IoT represents a network of devices, each equipped with a variety of sensors (and actuators) that sense and collect data about the local environment. For example, an IoT device may possess an array of sensors that monitors the air temperature, carbon monoxide level, wifi signals, and sound intensity. Data created by an IoT device goes through three main phases: the first phase is the initial creation, which takes place on the device, and then sent over the Internet to a central database system (e.g., the cloud). The second phase is how the central database system collects and organizes this data. The third phase is the ongoing use of such data by myriad applications, which include but are not limited to smart home, smart city, the industrial internet, connected cars, and connected health.

Data generated by IoT devices has the following key characteristics: *(i) Spatial and Temporal:* Data collected by a device represents a physical quantity, i.e., a signal, that continuously varies over time. For instance, an audio (sound) signal represents the variation of the sound intensity over the time dimension. Also, IoT devices are usually installed in a static location such as a building or a traffic intersection. However, some IoT devices are attached to objects that move in space like a connected vehicle or a wearable device. In both cases, the data generated by an IoT device possess a spatial location attribute that represents the longitude and latitude coordinates of the sensed observations. *(ii) Heterogeneous and Interconnected:* Not all IoT sensors are the same; a spectrum of various sensors exists and each sensor measures a different thing. For instance, a camera generates image signals while a microphone generates an audio signal; each signal type has different physical and mathematical characteristics [2]. For example, an ultrasonic motion sensor sends a signal when it detects motion whereas a microphone sensor only sends a signal when it detects a sound. Also, IoT data is inherently interconnected and can be linked to other data sources such as the city data, the web and social media.

Given such characteristics, a central data system must provide a spatial / spatio-temporal data management query Application Programming Interface (API) side-by-side with a digital signal processing API for programmers to develop IoT applications. Such requirement makes it really difficult for off-the-shelf systems to digest and process IoT data. The problem becomes even more challenging as the volume of data collected from IoT devices increases at a staggering rate. Today, RADAR, LIDAR, and CAMERA sensors have a bandwidth of up to 15 Mbit/sec, 100 Mbit/sec, and 3500 Mbit/sec, respectively. A connected vehicle, equipped with several of such sensors, generates data at a rate that exceeds 25 GB per hour [3]. The volume of such data will increase even more with already 2.8 trillion IoT devices deployed globally in homes and road intersections by the year 2019 [4], the 10 million self-driving cars equipped with dozens of sensors to be on the road and the 7 million drones to be flying in U.S. skies by 2020 [5, 6]. That makes it almost impossible for state-of-the-art big spatial data systems to handle real-time or near-real time IoT analytics applications.

2 A Spatio-Temporal Data Model for IoT Data

Recently proposed IoT data models [7, 8] over the semantics of things, sensor observations, and applications. Several works model IoT data for health-specific application such as real-time prediction of blood alcohol content using smartwatch sensor data [9]. Many papers model IoT data for smart city applications [10, 11]. One potential research direction will build upon these efforts. We can model IoT data using the `SenosorThings` standard published by the Open Geospatial Consortium (OGC) in 2016. The standard provides a formal / generic API to model and query IoT things, which can represent a smart car, smart watch, traffic camera, smart oven, etc... The API also models the collected observations, connections among IoT devices as well as their spatial and temporal attributes [12].

Recent works model IoT data, given its interconnected and linked nature, as a graph [13, 14], which integrates data generated by various IoT devices as well as integrates IoT data with other relevant data sources, e.g., semantic web and infrastructure networks. Thus, it makes sense to model linked IoT data as a graph and store it in a graph database system such as Neo4j [15] and Titan [16], which also support spatiotemporal attributes that can capture the location and time aspects of sensed observations [17, 18, 19, 18, 20, 21]. As depicted in Figure 1, the IoT graph is modeled as a property / labeled graph $G = (V, E, \varphi, \psi)$ such that (1) V is a set of vertexes that represent things (IoT devices and sensors) in an IoT network. Vertexes can also represent real world entities generated by other data sources, e.g., Knowledge Graphs, Social Graphs, and the Web. (2) E is a set of edges that connect IoT things and also connect IoT things to other entities collected from other data sources. (3) φ is a mapping function $\varphi : V \rightarrow \mathcal{L}$, where $\mathcal{L}$ is a set of labels or types, (4) ψ is a mapping function $\psi : E \rightarrow \mathcal{L}$ [22]. The IoT graph is a property graph that possesses at least one spatial label/type in $\mathcal{L}$. In such a graph, some vertexes semantically represent spatial objects, which possess spatial location attributes, e.g., point, polygon. The spatial attribute of a vertex is denoted as $v.loc$. Given such graph, a user may ask a query like "Find patterns in the IoT graph of audio signals observed by an IoT device located in Downtown Tempe such that the observation value is larger than 140 dB". A main challenge though is how a graph data system can evaluate the combination of spatial (e.g., range, K-Nearest Neighbors, spatial join), temporal, and graph predicates on linked IoT data [23, 24, 25].

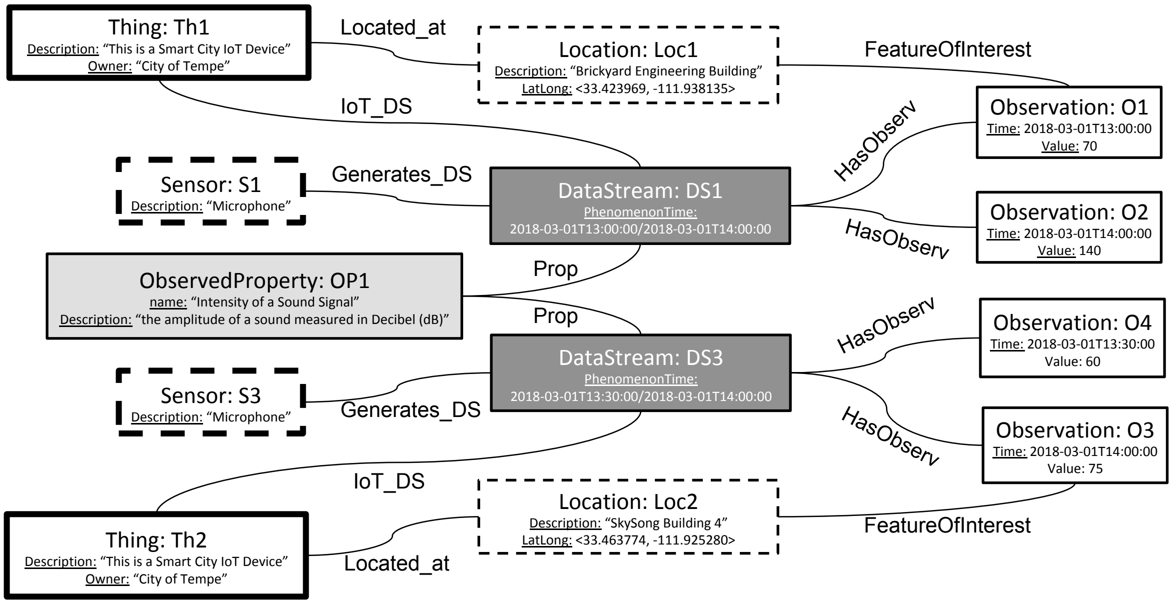

Figure 1: IoT modeled using the SensorThings OGC standard

3 Scalable Geospatial Processing of IoT data

There exist a few research efforts, which focus on developing scalable data infrastructure for IoT data [26, 9]. For instance, Dey et. al proposed a scheme where edge devices offer free computational slots to servers in a cloud based data analytics system [27]. The Aura system prototype [28] enables building ad-hoc clouds using IoT devices in the nearby physical environment. Jiang et. al evaluated cloud-based data storage options for IoT data [29] whereas Li et. al experimented the use of NoSQL database technology to store massive-scale IoT data [30]. However, none of these efforts takes into account spatio-temporal data processing operations, which are vital for processing IoT data [9].

State-of-the-art spatial and spatio-temporal data systems [31, 21, 32, 33, 34] do not provide native support for digital signal processing and machine learning operations whereas numerical frameworks such as MatLab do not provide in-house support for spatial data management. Futhermore, Such systems do not provide an out-of-the-box API to handle IoT sensor data. For instance, the Apache Spark `Dstream` API that can chop up live IoT data stream, represented as observations, into batches of X seconds. Spark can then treat each batch of data as an RDD and processes them using RDD operations. Finally, the processed results of the RDD operations are returned in batches. However, such APIs are not easy for programmers to develop IoT applications and are not natively optimized to run the combination of spatial, spatio-temporal, digital signal processing, and machine learning operations on IoT data. Examples of queries that require such combination include: "Q1: Report gunshots heard in Downtown Tempe area between 13:45 and 14:00 pm on 2018/03/01", "Q2: Report gunshots heard nearby (e.g., within 0.1 mi) a School in the City of Tempe". Q1 needs to filter out IoT data records that neither lie within Downtown Tempe nor observed between 13:45 and 14:00 pm on 2018/03/01. Q1 also needs to convert the audio signal into a time-frequency representation, e.g.., mel-spectrogram, by applying Fast Fourier Transform (FFT), then discrete cosine transform, and finally applying an urban sound classifier to detect gunshots. Q2 does the same with the exception that it applies a spatial join operation to retrieve observations only nearby schools.

Having said that, a promising research direction will incorporate IoT data awareness in state-of-the-art big spatial data systems such as Apached Sedona (GeoSpark) [35]. Furthermore, that also requires that system developers design a middleware framework, which understands the IoT devices streaming data to the central data system on one side and the requirements of applications accessing such IoT data on the other side. The proposed middleware system will tune the central data system to adaptively decide whether or not to eagerly propagate data from the device to the central system. I plan to modify existing relational and spatial query processing algorithms to leverage the IoT device capabilities and handle the different rate and types of data generated by various IoT devices. To capture the interconnected nature of IoT data, a spatial data system must also provide a graph processing API in addition to the spatial / spatio-temporal API. Such a combination is already supported by existing graph data systems [15, 16, 36, 37], however such systems treat the spatial attribute as a second class citizen, and hence cannot achieve real time or near real time performance. The community needs to craft efficient query operators that accelerate location-aware graph queries and also investigate new index structures that take into account network aspect of linked IoT data as well as the spatial and Spatio-Temporal aspects.

4 Conclusion

The Internet of Things (IoT) is getting more popular every day. The spatial, temporal, big, fast, heterogeneous, and interconnected nature of collected IoT data makes it difficult for off-the-shelf spatial data systems to digest and process such data, especially for real-time or near-real time analytics applications. The goal of this article is to encourage the community to design and develop spatial and spatiotemporal data infrastructure that can capture, store, query, analyze data from connected IoT devices at scale. The outcome of that research can provide a tool for data scientists, policy makers, and businesses to better utilize and extract value from IoT data growing at a staggering rate.

5 Acknowledgements

This work is supported by the National Science Foundation under award #1845789.

References

[1] $267 Billion will be spent on IoT technologies by 2020. https://www.forbes.com/sites/louiscolumbus/2017/01/29/internet-of-things-market-to-reach-267b-by-2020/#7ba38b21609b.

[2] A.V. Oppenheim, A.S. Willsky, and S.H. Nawab. *Signals and Systems*. Prentice Hall, 1997.

[3] Connected Vehicle Data. https://www.hitachivantara.com/en-us/pdf/white-paper/hitachi-white-paper-internet-on-wheels.pdf.

[4] 2.8 trillion sensor devices by 2019. https://www.bloomberg.com/news/articles/2013-08-05/trillions-of-smart-sensors-will-change-life-as-apps-have.

[5] 10 million self-driving cars will be on the road by 2020. http://www.businessinsider.com/report-10-million-self-driving-cars-will-be-on-the-road-by-2020-2015-5-6.

[6] 7 million drones flying in U.S skies by 2020. https://www.faa.gov/news/updates/?newsId=85227.

[7] Pratikkumar Desai, Amit Sheth, and Pramod Anantharam. Semantic gateway as a service architecture for iot interoperability. In *IEEE International Conference on Mobile Services (MS)*, pages 313–319. IEEE, 2015.

[8] Yong-Shin Kang, Il-Ha Park, Jongtae Rhee, and Yong-Han Lee. Mongodb-based repository design for iot-generated rfid/sensor big data. *IEEE Sensors Journal*, 16(2):485–497, 2016.

[9] Anne H Ngu, Mario Gutierrez, Vangelis Metsis, Surya Nepal, and Quan Z Sheng. Iot middleware: A survey on issues and enabling technologies. *IEEE Internet of Things Journal*, 4(1):1–20, 2017.

[10] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Sensing as a service model for smart cities supported by internet of things. *Transactions on Emerging Telecommunications Technologies*, 25(1):81–93, 2014.

[11] M Mazhar Rathore, Awais Ahmad, Anand Paul, and Seungmin Rho. Urban planning and building smart cities based on the internet of things using big data analytics. *Computer Networks*, 101:63–80, 2016.

[12] OGC SensorThings API. `http://docs.opengeospatial.org/is/15-078r6/15-078r6.html#17`.

[13] Payam M. Barnaghi and Mirko Presser. Publishing linked sensor data. In *Proceedings of the 3rd International Workshop on Semantic Sensor Networks, SSN 2010, Shanghai, China, November 7, 2010*, 2010.

[14] Laurent Lefort, Josh Bobruk, Armin Haller, Kerry Taylor, and Andrew Woolf. A linked sensor data cube for a 100 year homogenised daily temperature dataset. In *Proceedings of the 5th International Workshop on Semantic Sensor Networks, SSN12, Boston, Massachusetts, USA, November 12, 2012*, pages 1–16, 2012.

[15] Neo4j graph database. `https://neo4j.com/`.

[16] Titan distributed graph database. `http://titan.thinkaurelius.com/`.

[17] Petko Bakalov, Erik G. Hoel, and Sangho Kim. A network model for the utility domain. In *SIGSPATIAL/GIS*, pages 32:1–32:10. ACM, 2017.

[18] Mohamed Sarwat, Jie Bao, Chi-Yin Chow, Justin J. Levandoski, Amr Magdy, and Mohamed F. Mokbel. Context awareness in mobile systems. In *Data Management in Pervasive Systems*, pages 257–287. 2015.

[19] Kyriakos Mouratidis, Jing Li, Yu Tang, and Nikos Mamoulis. Joint search by social and spatial proximity. *TKDE*, 27(3):781–793, 2015.

[20] Yuhan Sun, Nitin Pasumarthy, and Mohamed Sarwat. On Evaluating Social Proximity-Aware Spatial Range Queries. In *Proceedings of the International Conference on Mobile Data Management, MDM*, 2017.

[21] Mohamed Sarwat. Interactive and Scalable Exploration of Big Spatial Data–A Data Management Perspective. In *Proceedings of the International Conference on Mobile Data Management, MDM*, 2015.

[22] Wen Sun, Achille Fokoue, Kavitha Srinivas, Anastasios Kementsietsidis, Gang Hu, and Guo Tong Xie. Sqlgraph: An efficient relational-based property graph store. In *SIGMOD*, pages 1887–1901, 2015.

[23] Mohamed Sarwat and Yuhan Sun. Answering Location-Aware Graph Reachability Queries on GeoSocial Data. In *Proceedings of the International Conference on Data Engineering, ICDE*, 2017.

[24] Yuhan Sun and Mohamed Sarwat. A Generic Database Indexing Framework for Large-Scale Geographic Knowledge Graphs. In *Proceedings of the ACM Symposium on Advances in Geographic Information Systems, ACM GIS*, 2018.

[25] Yuhan Sun and Mohamed Sarwat. A spatially-pruned vertex expansion operator in the neo4j graph database system. *GeoInformatica*, 23(3):397–423, 2019.

[26] Joshua Cooper and Anne James. Challenges for database management in the internet of things. *IETE Technical Review*, 26(5):320–329, 2009.

[27] Swarnava Dey, Arijit Mukherjee, Himadri Sekhar Paul, and Arpan Pal. Challenges of using edge devices in iot computation grids. In *International Conference on Parallel and Distributed Systems (ICPADS)*, pages 564–569. IEEE, 2013.

[28] Ragib Hasan, Md Mahmud Hossain, and Rasib Khan. Aura: an iot based cloud infrastructure for localized mobile computation outsourcing. In *IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, pages 183–188. IEEE, 2015.

[29] Lihong Jiang, Li Da Xu, Hongming Cai, Zuhai Jiang, Fenglin Bu, and Boyi Xu. An iot-oriented data storage framework in cloud computing platform. *IEEE Transactions on Industrial Informatics*, 10(2):1443–1451, 2014.

[30] Tingli Li, Yang Liu, Ye Tian, Shuo Shen, and Wei Mao. A storage solution for massive iot data based on nosql. In *IEEE International Conference on Green Computing and Communications (GreenCom)*, pages 50–57. IEEE, 2012.

[31] Ram Sriharsha. Geospatial analytics using spark. `https://github.com/harsha2010/magellan`.

[32] Dong Xie, Feifei Li, Bin Yao, Gefei Li, Liang Zhou, and Minyi Guo. Simba: Efficient In-Memory Spatial Analytics. In *Proceedings of the ACM International Conference on Management of Data, SIGMOD*, 2016.

[33] Jia Yu, Jinxuan Wu, and Mohamed Sarwat. A demonstration of GeoSpark: A cluster computing framework for processing big spatial data. In *Proceedings of the International Conference on Data Engineering, ICDE*, 2016.

[34] Jia Yu, Zongsi Zhang, and Mohamed Sarwat. Spatial data management in apache spark: the geospark perspective and beyond. *GeoInformatica*, 23(1):37–78, 2019.

[35] Apached Sedona. `http://sedona.apache.org`.

[36] James Cheng, Silu Huang, Huanhuan Wu, and Ada Wai-Chee Fu. Tf-label: a topological-folding labeling scheme for reachability querying in a large graph. In *SIGMOD*, pages 193–204. ACM, 2013.

[37] Andy Diwen Zhu, Wenqing Lin, Sibo Wang, and Xiaokui Xiao. Reachability queries on large dynamic graphs: a total order approach. In *SIGMOD*, pages 1323–1334. ACM, 2014.