Processing Moving Object Data Streams with Data Stream Management Systems

Tobias Brandt
Supervised by Marco Grawunder
University of Oldenburg, Germany
tobias.leo.brandt@uol.de

ABSTRACT

With the wide spread of cheap and mobile GPS sensors as well as mobile data connections, live streams from moving objects are becoming a huge data source. The services based on these data streams, for example, for connected cars, vessels or smartphone users, need real-time results for queries based on the current or even near-future positions of the moving objects. Spatio-temporal data from moving objects cannot just be treated as a crowd of points with timestamps, but must be seen as points in a trajectory with non-measured points in between. In this paper I present my work on the management of such real-time trajectories within a Data Stream Management System (DSMS) to enable simple, flexible and efficient in-memory moving object query processing.

1. INTRODUCTION

Moving objects in the real world are everywhere and have been there for a while: pedestrians and cars on the streets, vessels on the oceans and airplanes in the sky. Comparably new is the huge amount of data these objects are producing. Many of these send their location regularly to a central server or other facility, may it be the smartphone user with a Location-based Service (LBS) or a vessel with an Automatic Identification System (AIS) sender.

This data can be used to answer questions and solve real-world problems. For example, vessels can be warned about congested or currently dangerous areas based on their own position as well as the positions of other vessels. As more data can be shared via live-streams and as the results of such queries are required with minimal delay, traditional systems that first store and then query the data streams are not ideal. They typically run short-term queries on static data sets that are stored on the hard drive.

Data Stream Management Systems (DSMSs) especially target data streams. They offer solutions for many data stream related challenges, provide query languages to define queries without the need to write code in a general purpose programming language and simplify the connection to typical data sources. Maintenance of queries on data streams is made simple due to the ease to change and update the query text. Hence, queries can be adapted to new requirements quickly. These features make them a useful tool to easily create and change queries for many different use cases and are therefore a good choice for rapid prototyping systems in the field of data stream processing and analysis.

To cope with the requirements of data streams, DSMSs support continuous queries and use a data-driven approach. New results are incrementally calculated when new data arrives at the system. This increases the demand for quick calculations, wherefore data is typically kept in-memory. Unfortunately, a potentially infinite data stream cannot be hold in-memory. A typical solution DSMSs provide are windows. These reduce the amount of data hold in-memory to a smaller part of the data streams, e. g., all elements from the last hour or the last 100 elements.

Even though DSMSs already tackle lots of the challenges that occur with data stream processing, they lack features necessary to work with moving objects data. Two of these features are (1) continuous location interpolation and near-future prediction and (2) fast moving objects index structures for windows with high fluctuation. In this work, I concentrate on point data, hence, moving regions are not in the scope. That is because most moving object data of interest today, such as vessels and pedestrians, can be simplified to point objects without loosing too much precision in the queries. Moving or evolving regions in contrast introduce a whole new palette of challenges.

One important feature of moving objects data streams is that the objects move continuously but are only measured once in a while. Therefore, the objects have unknown locations in between the measurements, which can and sometimes need to be used for querying. Imagine a query where a vessel needs to know all vessels around it. Another vessel, which last known location is (temporal and spatial) far away but will probably be within that range on querytime, should be included in the answer, hence, its location needs to be automatically predicted to the future by the query. This scenario is particularly important for satellite AIS where it is normal to have hours between location updates [3]. Streaming data differs from static time-series data: in static data, all measured locations of a moving object are known. In contrast, in a streaming environment, it is not possible to know if and when the next location update of a moving object will arrive.

Challenges with this approach are that interpolation and

Proceedings of the VLDB 2017 PhD Workshop, August 28, 2017. Munich, Germany.

prediction depends on the use case and always comes with an uncertainty. When new data about an interpolated object is available, old query results may need to be updated. When and how to update results by more precise ones is a non-trivial question within a DSMS. To my best knowledge, there is no work that tackles these challenges in the field of data streams for moving objects.

The second feature mentioned above are moving objects windows and suitable index structures. As not all data can be stored, it needs to be decided which data is still needed for processing and which is not and how and when old data can be wiped. Typical window concepts such as time-based windows can be extended by windows especially for moving objects. A possible window type could be a distance-based window. It would store all data within a certain distance of a single moving object, e. g., the last kilometer of every object. The requirements for the underlying index structures differ both from pure temporal as well as pure spatial index structures.

2. OBJECTIVES AND CHALLENGES

The overall goal of this work is to create and implement concepts to allow DSMSs to process spatio-temporal data from moving objects. To reach this goal, I am tackling the challenges of including location interpolation and prediction as well as window concepts for moving object data streams.

2.1 Objectives

- **Location Interpolation** Moving objects naturally move continuously but are only measured once in a while. Within a data stream, they seem to be hopping from point to point, resulting in delayed and possibly wrong results. I aim to introduce location interpolation into the data stream processing to that the objects move continuously in time and space. The interpolation should also be used for short-term location prediction.

- **Result Uncertainty** As prediction always comes with a degree of uncertainty, the results are uncertain as well and can change when new data from a moving objects arrives. The accuracy or uncertainty of a result should be transparent to the user and updates of results should be possible.

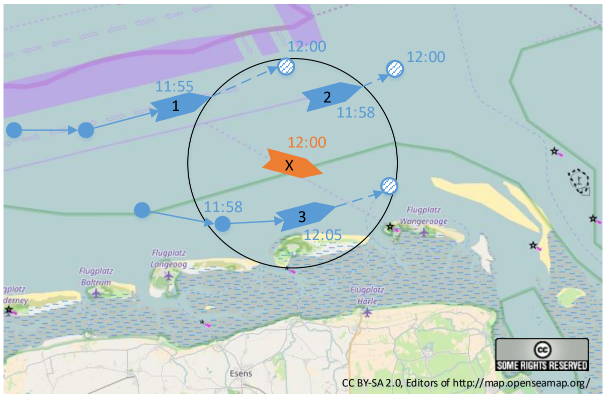
- **Moving Object Windows** Window definitions that are especially useful for moving objects data should be introduced.

- **Spatio-temporal Indexes** The data within the windows need to be hold in spatio-temporal index structures optimized for high data fluctuation.

The concepts created to solve the goals should be evaluated by implementing them into an open source DSMS. Scenarios with AIS data from vessels should show that the concepts work with real-world data and queries.

2.2 Challenges for Location Interpolation

Typical spatio-temporal queries include neighborhood and range queries. An example query could be “Continuously report all vessels within a range of 10 km around vessel X”. Such a query is depicted in Figure 1. The orange vessel in the middle sent its last location at 12 o’clock and wants to know which vessels are in its range at that point in time. The trajectory of the other vessels one to three are visualized with the arrows and circles. The circles are the measurements where the correct location of the vessel is known. The arrows in between visualize a simple interpolation. It is assumed that the path between the measurements is straight. That is not necessarily the case but it is a reasonable and simple approximation. The dashed lines with the striped circles are predictions of the future trajectory.



CC BY-SA 2.0, Editors of http://map.openseamap.org/

Figure 1: Range query for the orange vessel.

The need to interpolate and predict locations arises from the fact that the moving objects do not measure and send their location synchronized with each other, but at different time intervals. In the figure, the last known location of the orange vessel in the middle was captured at 12 o’clock. For this point in time, the locations of all other vessels are needed to answer the query correctly. Unfortunately, the known, i. e., measured, locations of the other vessels are not at 12 o’clock, but slightly before or after this point in time. If only the last known location would be used to answer the query, the result would be wrong: Vessel 1 would not be within the result but should be, Vessel 2 would be within the result but should not and the location of Vessel 3 would be wrong.

Hence, interpolation and prediction is necessary to answer the query approximately correct. When doing so, a few challenges arise. The interpolation has to work in an incremental manner with limited knowledge about the data, as not all past data can be stored in a streaming environment. The accuracy of a query result needs to be known to the user or further processing steps. When the prediction was wrong, old results for a query may need to be updated (e. g., if Vessel 1 takes a different path than predicted). In a streaming scenario, the approximate result may already be used for further processing or a following result, for example, for 12:03, has already been processed. The questions on how to integrate uncertain results and updates to them within a DSMS need to be tackled.

2.3 Challenges for Moving Object Windows

Windows are necessary to reduce the infinite data stream to a finite set of data. The data within a window can be kept in-memory and never needs to be permanently written to a hard drive. Next to the performance improvements, the concept of windows has more benefits. They are based on the assumption that in many cases, queries are only or mostly interested in the current data and not in the data from the distant past. To define a window according to the use case, the domain needs to be known. A typical window could, for example, hold all data from the last hour.

In the domain of moving objects, the requirements for windows can differ depending on the scenario. The definition of a window only by the time and not by the space dimension is often not enough. Imagine, for example, a window where the speed of all objects within a data stream is diverse and variable (e. g., slow vessels and fast planes within one stream). It could be necessary to have from each object at least the last kilometer within the window. With a time-based window this would be difficult to achieve. Additionally, compression could be introduced to moving objects windows. Patroumpas et al. [7] show that trajectory data can be compressed without loosing much accuracy. Hence, new window concepts for moving objects are useful.

Windows reduce the amount of the required (in-memory) storage space. Nevertheless, it opens up new questions about how to clean up the memory, for example, when to delete old data. This question gets more complicated as window indexes have to be shared between multiple queries. Imagine a data stream of all vessels on the North Sea. As spatial queries can be heavily accelerated when using an index, the data in the windows are indexed. Thereby, due to memory limitations, creating multiple nearly identical indexes must be avoided. Subsequently, one index is shared between multiple queries. Nevertheless, such a sharing makes the decision when to delete old data more complicated, as the window requirements from the queries can be different.

Additionally, not every index structure is suitable for a spatio-temporal data stream index. In contrast to more traditional Geographic Information System (GIS) applications, the fluctuation in the data is very high. New data needs to be inserted and at the same time old data needs to be removed on a high frequency. It is possible that the whole set of data within a window can be swapped within minutes or even seconds, which, for example, distinguishes the requirements from static time-series data. Traditional index-structures that require heavy reorganization when data gets changed are probably not suitable for this environment. In this work it needs to be evaluated if index structures for this purpose are useful as it is only an improvement if the indexing needs less time than it saves while querying the data.

In this PhD project I aim to create suitable window concepts, implement them and choosing an efficient index structure for this very dynamic environment.

3. RESEARCH PLAN

In this section, the main approaches to overcome the challenges from above are described.

3.1 Approach

Query processing on data streams is typically done with operator graphs. The elements of the data stream are send from one operator to the next, each operator doing a specific task. Joins, projections and selections are typical examples for such operators.

When adding support for moving objects data, this modular architecture should be exploited. New operators can implement the spatio-temporal operations. While doing so, they have to behave like normal operators to the outside so that other operators can seamlessly use the output. Two example operators that are needed are a range and a k-nearest neighbors (kNN) operator. Both search for other moving objects close to a certain object. The external behavior of these operators is similar to other operators. They receive stream elements, process them and send their results as stream elements to the next operator.

Internally, these operators need to use location interpolation and prediction to compute correct results and annotate these results with the level of (un)certainty in the meta data of a streaming object. The interpolation should be done by a framework within the DSMS that allows to interchange algorithms, as the interpolation algorithm can change from case to case.

3.2 Current and Future Work

Currently, prototype versions for moving object range and kNN queries are available. A prototype implementation within a DSMS was developed. For spatial querying with moving object windows, an index structure based on GeoHashes [6] was developed. It showed better performance than an implementation based on QuadTrees. This could be due to better insertion and deletion performance with the GeoHash implementation. However, these results are very preliminary, as the test setup needs to be better described and results need to be analyzed further to find out the reasons for the differences.

The correct integration of those queries as well as moving object windows is ongoing work. The results are currently only partly usable for other query operators. Interpolating and predicting locations are in the concept phase. Development and implementation of these will be a major part of the future work.

3.3 Planned Evaluation

The concepts that are created in this PhD project will be implemented into the open source DSMS *Odysseus*[1] [1]. Odysseus offers a rich set of operators and a query language. For the purpose of this work it already supports protocols used in the maritime domain such as AIS and can talk to common data sources such as RabbitMQ out of the box. In contrast to streaming frameworks such as Apache Flink[2] or Heron[3], it is not necessary to program in a general programming language such as Java to create new queries.

With that implementation, the feasibility of the concepts will be evaluated. Using an iterative approach, the concepts can be adjusted if uncovered challenges occur while evaluating. The implementation will be used with scenarios in the maritime context, especially with AIS data. An example query could be to continuously query if a vessel is heading to an area that will be congested during its transit.

For the given scenarios with moving objects, timely query results are necessary, wherefore the performance of the solutions will be measured. The latency and throughput of the queries will be used to compare different implementations, e. g., different approaches for spatio-temporal indexes.

[1] `http://odysseus.offis.uni-oldenburg.de/`, last accessed on 03/21/2017

[2] `https://flink.apache.org/`, last accessed on 05/24/2017

[3] `https://twitter.github.io/heron/`, last accessed on 05/24/2017

4. RELATED WORK

General purpose and open source streaming systems such as Apache Flink and Apache Storm[4] as well as commercial systems such as IBM InfoSphere Streams[5] (short: IBM Streams) offer high performance and distributed stream processing, but have only limited support for moving objects. While spatio-temporal data is supported in some systems (e. g., with IBM Streams [2]), location interpolation and moving object windows are, as of my knowledge, not.

Zhang et al. [9] use Apache Storm to process fast data streams from moving objects. They focus on a distributed spatial index which speeds up range and kNN queries as well as spatial joins. One main difference to this work is that they do not interpolate and predict locations to have temporal correct results.

The open source project GeoMesa[6] works with spatio-temporal data, e. g., from moving objects [6]. The project develops indexes based on space-filling curves. These allow quick access to spatial or spatio-temporal data within sorted key-value stores such as Apache Accumolo[7]. GeoMesa does not specifically address streaming data, data stream management capabilities or location interpolation.

The RxSpatial library [8] is an extension for the Microsoft SQL Server Spatial Library and adds support for moving objects. It adds the RUM-tree, which is an extension of the R-tree for frequent updates. Additionally, RxSpatial allows continuous spatial queries, e. g., to observe if a moving object is close to another. As of my best knowledge, the library does not take into account the time of the updates but uses the newest updates of every moving object. Interpolation and prediction are not used.

Interpolation for moving objects is a difficult challenge for moving regions (e. g., described by Heinz et al. [5]). This is especially complex as these regions can change their shape over time. In this work I want to concentrate on moving points, which is a way simpler version of that problem. Nevertheless, the perfect interpolation method is not the goal of this work but the integration of interpolation and prediction into the stream processing of moving objects data.

Secondo [4] is a database system especially for moving objects. It has a spatial and temporal algebra with which queries for moving objects can be formulated. As it is a database, it is not optimized for data streams, e. g., it does not support windows, does not run mainly in-memory and hence does not have to solve the problem of cleaning up old data. Nevertheless, it gives useful insights on the handling of moving objects data.

Patroumpas et al. [7] use AIS data in a streaming environment to detect complex events such as unexpected stops of vessels. They compress the data to important points in the trajectory of the vessels without loosing much accuracy. They also address errors in the AIS data by removing wrong measurements. In contrast to this work, they do not use location prediction for vessels that did not send an update for a while, which is for example the case for satellite AIS.

[4] https://storm.apache.org/, last accessed on 03/21/2017

[5] https://www.ibm.com/analytics/us/en/technology/stream-computing/, last accessed on 03/21/2017

[6] http://www.geomesa.org, last accessed on 03/17/2017

[7] https://accumulo.apache.org/, last accessed on 03/21/2017

5. CONCLUSION

This paper describes the motivation, challenges and approaches of processing data from moving objects in DSMSs. A major challenge are asynchronous updates of locations of multiple moving objects. To serve timely query results, e. g., for a range query, locations of objects need to be interpolated and predicted. The integration of interpolated values into a DSMS provides some challenges that are tackled with this PhD project. First approaches to solve these are explained. The planned evaluation uses AIS data from vessels for continuous queries.

6. ACKNOWLEDGMENTS

We thank the Ministry of Science and Culture of Lower Saxony, Germany for supporting us with the graduate school *Safe Automation of Maritime Systems (SAMS)*.

7. REFERENCES

[1] H.-J. Appelrath, D. Geesen, M. Grawunder, T. Michelsen, and D. Nicklas. Odysseus: A highly customizable framework for creating efficient event stream management systems. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, DEBS '12, pages 367–368, New York, NY, USA, 2012. ACM.

[2] A. Biem, E. Bouillet, H. Feng, A. Ranganathan, A. Riabov, O. Verscheure, H. Koutsopoulos, and C. Moran. Ibm infosphere streams for scalable, real-time, intelligent transportation services. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 1093–1104, New York, NY, USA, 2010. ACM.

[3] M. A. Cervera, A. Ginesi, and K. Eckstein. Satellite-based vessel automatic identification system: A feasibility and performance analysis. *International Journal of Satellite Communications and Networking*, 29(2):117–142, 2011.

[4] R. H. Güting, T. Behr, and C. Düntgen. *Secondo: A platform for moving objects database research and for publishing and integrating research implementations.* Fernuniv., Fak. für Mathematik u. Informatik, 2010.

[5] F. Heinz and R. H. Güting. Robust high-quality interpolation of regions to moving regions. *Geoinformatica*, 20(3):385–413, July 2016.

[6] H. V. Le. Distributed moving objects database based on key-value stores. In *Proceedings of the VLDB 2016 PhD Workshop co-located with the 42nd International Conference on Very Large Databases (VLDB 2016), New Delhi, India, September 9, 2016.*, 2016.

[7] K. Patroumpas, E. Alevizos, A. Artikis, M. Vodas, N. Pelekis, and Y. Theodoridis. Online event recognition from moving vessel trajectories. *GeoInformatica*, 21(2):389–427, 2017.

[8] Y. Shi, A. M. Hendawi, H. Fattah, and M. Ali. Rxspatial: Reactive spatial library for real-time location tracking and processing. In *Proceedings of the 2016 International Conference on Management of Data*, pages 2165–2168. ACM, 2016.

[9] F. Zhang, Y. Zheng, D. Xu, Z. Du, Y. Wang, R. Liu, and X. Ye. Real-time spatial queries for moving objects using storm topology. *ISPRS International Journal of Geo-Information*, 5(10), 2016.