Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 731–736

Procedia Computer Science

www.elsevier.com/locate/procedia

The 8th International Conference on Ambient Systems, Networks and Technologies (ANT 2017)

# STEAM: A Platform for Scalable Spatiotemporal Analytics

Bersant Deva[a,*], Philip Raschke[a], Sandro Rodriguez Garzon[a], Axel Küpper[a]

[a]*Service-centric Networking, Telekom Innovation Laboratories, Technische Universität Berlin, Germany*

**Abstract**

Spatiotemporal datasets have become increasingly available with the introduction of a various set of applications and services tracing the behavior of moving objects. Recently, there has been a high demand in understanding these datasets using spatiotemporal analytics. While being considered of high value, spatiotemporal analytics did not yet see a wide spreading into the actual business workflow or the direct configuration of services and applications. The computational complexity for spatiotemporal datasets and the heterogeneity of data sources are considered key factors for the current state. This paper introduces STEAM, a platform for distributed spatiotemporal analytics on heterogeneous spatiotemporal datasets. STEAM introduces a framework that abstracts the key components from incoming spatiotemporal datasets that originate from various positioning systems. This abstraction provides a common base for distributed and scalable analytics methods that is not bound to a specific underlying positioning technique. STEAM provides a distributed state-of-the-art implementation and is evaluated on a multi-machine testbed for linear scalability.


Peer-review under responsibility of the Conference Program Chairs.

*Keywords:* Big Data Applications; Spatiotemporal Analytics; Service-oriented Architectures.

## 1. Introduction

With the rise of mobile and ubiquitous computing an ever increasing amount of spatiotemporal data is available. This data originates from a various set of sources, such as location-based services (LBS) applications, fleet GPS tracking systems or mobile and wireless networks. In recent times, the introduction of Internet-of-Things technology has further increased the set of devices, which either emit their own location along with other data, such as connected environmental sensors or track other objects' location using motion detectors, sensors or Bluetooth beacons. Furthermore, other frequently moving objects get increasingly connected such as cars, delivery trucks, buses, trains or more recently robots, drones and other autonomous vehicles. With this growing set of traceable devices, different positioning techniques and the expanding data resources for several domain areas, the volume and heterogeneity of spatiotemporal data evolved into a challenging task to manage. It has become crucial to sanitize, filter, derive and visualize the data in order to extract useful and perceptive information accordingly. Due to the characteristics of spatiotemporal data, this process, known as location, movement or spatiotemporal analytics, is usually individually adapted by experts

* Corresponding author. Tel.: +49-30-8353-58676
*E-mail address:* bersant.deva@tu-berlin.de


Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.05.429

with domain knowledge for each dataset, system and environment. However, most of these spatiotemporal analytics approaches provide only the computation and visualization of previously strictly defined queries, e.g. kernel density estimation[1], trajectory estimations[2] or density-based clustering[3]. Spatiotemporal processing describes the process of transforming spatiotemporal data to spatiotemporal statistics and movement patterns, referred as spatiotemporal insights throughout the paper. This process is considered as highly computational complex and therefore problematic in terms of scalability, even for well-known and rather simple point-in-polygon queries. While many approaches exist that adapt to the characteristics and complexity of spatiotemporal data by parallelization and distribution of tasks[4], there has been only little effort to make the resulting spatiotemporal insights of this process available for further processing by third party services. These could be used, for example, directly in LBS and applications. Additionally, far-reaching spatiotemporal patterns can be detected that only become visible over a period of time, e.g. reoccurring spatiotemporal events in certain time intervals. For the purpose of convenient, reliable and scalable serviceability, the overall process of spatiotemporal processing requires some sort of harmonization with a common output interface.

For this reason, STEAM, a platform for scalable spatiotemporal analytics is introduced. STEAM allows the input of several heterogeneous data sources by adapting and mapping incoming data according to a defined taxonomy in parallel. The analytics computation is carried out in a distributed and scalable manner by using *Apache Spark*. The STEAM analysis process is evaluated to assess linear scalability using the spatiotemporal NYC TLC dataset[5], which contains taxi movement history for New York City. The output of the STEAM analysis process is provided through an application programming interface (API) to allow for convenient querying of data, which complies with the dimension and metric paradigm previously introduced for location analytics[6]. This way, the insights can be directly queried for further use with more advanced analytics and LBS applications.

## 2. Related Work

Spatiotemporal analytics is not a new topic and issues with the scalability of spatiotemporal processing are well known and highly targeted in the research community, from the point-in-polygon problem[7] to the distribution of R-Trees[8] in order to be able to query large datasets on distributed systems. Spatiotemporal analytics methods are manifold and applied to several domain areas[9]. Taxis[10], mobile device users[11] or car sharing rides[12] have been analyzed extensively over the last couple of years. While most of the related work focuses on the analysis of GPS data with WGS84 coordinates[13,10,12,14,15,16], also other positioning techniques based on radio frequency (RF) signals gained some attention, such as cellular network[11] or WiFi data[2,6]. These approaches are often used when the reliability of global navigation satellite system (GNSS) positioning technologies cannot be guaranteed or the monitoring of movement is not applicable on mobile devices. The common ground with all related work is that spatiotemporal analytics is considered computationally complex and requires a scalable way of computation. Aggregation methods, such as the well known Map-Reduce, have been already applied for density-based clustering[4] or spatiotemporal statistics[6] to provide knowledge on movement behavior. Also, more advanced general-purpose distributed computing frameworks, for example Apache Spark, have been used in spatiotemporal analytics for hotspot detection[14] or GPS accuracy aggregation[16] respectively. Spatiotemporal analytics has been used in conjunction with (Web) services and is, as such, not a new idea. The approach of a service-based workflow for scientific geospatial processing was first introduced by Jaeger et al.[13]. However, this has only recently gained the interest of researchers as the data amount is growing and more appropriate technology for scalable analytics becomes available. Only few approaches[6,12,17] mention the consideration of their analytics insights to be applied in LBS applications. Currently, the only available approach that uses spatiotemporal analytics on historical data to enhance its platform is *CATLES*[16], a 3D world simulator for the evaluation of LBS. Based on the current position in the simulator the interface shows the GPS accuracy that can be expected in this area. This information is used by developers to evaluate their LBS applications in different urban or rural environments. Noteworthy, the often available *kNN*-Search of existing LBS has already been used to apply spatiotemporal analytics and gain knowledge in the *ANALOC* platform[15].

## 3. Concept

Most of the currently known platforms and frameworks for spatiotemporal analytics target the creation and adaptation of algorithms to solve previously defined queries on specific spatiotemporal datasets. These platforms focus

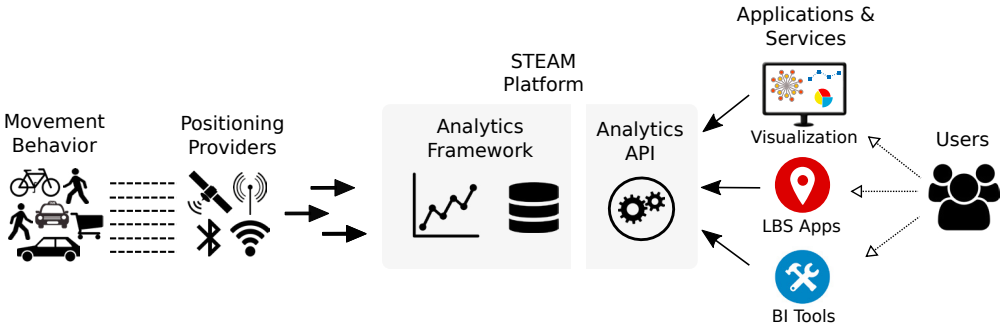


Fig. 1. Workflow of the STEAM ecosystem: From movement behavior to service availability and use

mostly on a particular positioning system, such as GNSS or RF signaling techniques. In contrast, STEAM focuses on providing analytics for all kind of spatiotemporal data by abstracting on its key components into processable entities. The STEAM platform allows multiple sources of spatiotemporal data input, and generates analytics insights, which are accessible through a common application interface. With the expanding availability of positioning systems outdoors and indoors, the movement behavior of objects, whether it is cars, bikes, shopping carts, or mobile devices is increasingly traced. While positioning systems and analyzed devices vary, the essential queries for spatiotemporal analytics maintain the same and basically reflect the questions *what* happened with *whom*, *where*, and *when*. These translate directly with the *key components* of spatiotemporal data, which are *object*, *location* and *time*. This is true for a variety of spatiotemporal analytics use cases, whether examining the traffic congestion in a large city, looking at dwell times of customers in a shopping mall or when trying to predict the incoming amount of passengers in an airport to optimize security lanes. The major difference, however, is the availability and reliability of data for this variety of tasks, which can be very heterogeneous with different positioning systems as a result of adaptation to specific surrounding requirements. For example, GNSS have proven to work sufficiently precise in open-space outdoor environments, but lack accuracy indoors and in cities with large buildings. The real-world applicability often defines the usage of a certain position technique over another. Thus, as there is no ”one size fits all”-solution for positioning techniques, spatiotemporal analytics systems are required to adapt to these. Nevertheless, this does not necessarily mean that analytics methods require to be bounded to each positioning technique, as the data ground provided, stays the same with the three key components. The STEAM ecosystem, as depicted in Figure 1, considers multiple positioning providers that track the movement behavior of objects and serve as an input for the STEAM platform. This platform contains mainly two crucial components, the STEAM Analytics Framework, responsible for spatiotemporal analytics computation and the STEAM Analytics API, used to serve the computed insights to third party applications and services in form a well-defined programming interface. At first, insights could be served traditionally by visualizing the findings for interested users, or to critically influence decision-making processes. Another important aspect, not fully targeted yet, is the use of spatiotemporal insights for the modification of LBS applications or other context-aware services that consider previous movement behavior.

*STEAM Analytics Framework*

The STEAM Analytics Framework relies on the abstraction of positioning data based on the notion of space and a topological model to describe spatial relations in order to provide spatiotemporal metrics applicable to different positioning providers. Distance metrics are well known from Euclidean in n-dimensional space to the haversine formula, or can be inferred implicitly from attenuated radio frequency signal strength. The framework uses these measures to adapt to various incoming spatiotemporal data from different positioning providers. Additionally, the time component plays a crucial role in defining metrics for spatiotemporal analytics, especially when looking at the notion of movement for uniquely identifiable objects, but also when there is interest in an insights metric’s distribution over time. While basic occurrences can be measured without an *object* component, it is not possible to track movement over time if the object emitting the data is not identifiable. It is crucial to state, that the extraction of knowledge by spatiotemporal analytics is not limited by the underlying positioning techniques, but rather by the availability and quality of the key components of the incoming data. While being regarded as highly useful and promising for

the above mentioned domain areas, the applicability of scalable spatiotemporal insights to actual service provision has been lacking or was mostly limited to specific scenarios. As already mentioned, the computational complexity and the large size of spatiotemporal datasets challenge the provision of services based on these insights. Therefore, the execution can only be targeted by parallelization and distribution of tasks in a meaningful manner[4,6,14], which is another key requirement for the framework. With increasing complexity of algorithms and larger datasets, the aggregation and distribution of insights on a time-sliced basis will become necessary in order to be able to grasp movement patterns over time.

*STEAM Analytics API*

In order to analyze spatiotemporal data with the STEAM platform it is required to configure the STEAM Analytics Framework accordingly. Thus, the STEAM Analytics API allows the definition of regions of interest (ROI) which can be later on queried for analytics insights. The raw data is mapped as an abstraction between the location dimension used by the incoming datasets and the actual declared ROI. Furthermore, dimensions related to specific domains available in the incoming dataset can be defined within the API, which enhance the depth of gained knowledge from spatiotemporal insights. Metrics describing the movement between two regions could, for example, include the distribution of demographics of users, or the tip amount spent on a taxi ride. Additionally, the STEAM Analytics API allows external services to query the computed insights based on a set of dimensions and metrics related to stationary and movement analytics.

## 4. Proof of concept

In order to prove the applicability of the approach, the STEAM platform has been implemented and evaluated for linear scalability with different dataset sizes of New York City taxi rides[5] and a variation of analytics computing clusters. The key challenges for the implementation of the STEAM Analytics Framework are the following: (i) maintaining scalability with increasing data size, (ii) dealing with heterogeneous data, and (iii) the spatiotemporal complexity of the data. Figure 2 illustrates the data flow and the computation phases.

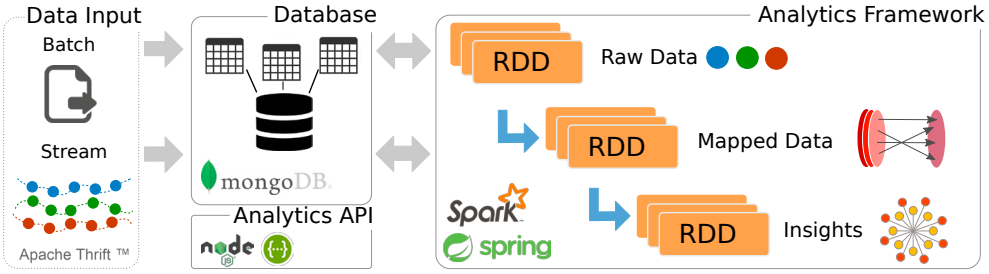


Fig. 2. Individual components and data flow of the STEAM platform implementation

The STEAM platform uses *MongoDB* to import raw spatiotemporal data as it suits best for storing heterogeneous data due to its flexible document structure. A cluster of MongoDB nodes can be created to distribute data and thus work load to enable potential for improved scalability. As a major component for distributed computing, Apache Spark has been used to ensure scalability with growing data size. *Apache Spring* is employed to structure the code base. The STEAM Analytics Framework connects to the MongoDB using the *MongoHadoop Connector*, allowing data to be loaded when required in a distributed manner. The raw data is parsed by the framework for the relevant key components, such as location and time of incoming spatiotemporal data entries. The regions of interest are represented by a tree data structure, which help to optimize the mapping process. A node in this tree is called *view*. Each view in the tree has a representation depending on the spatiotemporal data input used, for example geographical in form of one or multiple polygons. Afterwards, the parsed data is processed in two phases. In the first phase, input data is mapped to a leaf in the tree (point-in-polygon problem). If a position can be identified as contained in the considered view, the data entry is mapped to the leaf and all its ancestors. The spatiotemporal complexity of this mapping is

tackled by distributing its execution. In the second phase, the results are aggregated over common time periods, for example on a per hour, day, week, quarter, and year basis to represent the obtained insights in a convenient way. The data is completely processed in one iteration. The obtained insights are written back to the MongoDB, in order to be accessed through the STEAM Analytics API realized according to REST in *NodeJS*. The STEAM Analytics Framework was tested successfully with 2D Bluetooth based indoor and RF signaling (WiFi, mobile network) data[6] as well as geographical datasets, such as taxi data used in the following.

The evaluation is expected to yield insights about the framework's handling of large data quantities and its capabilities to tackle these with a bigger computation cluster. According to Jogalekar and Woodside distributed systems must be scalable, should be deployable in a wide range of scales, and an increase of capacity should be in proportion to costs while still maintaining the same quality of service[18]. The computation time required to process a certain amount of data with a specified number of worker nodes is measured. The published data of the NYC Taxi and Limousine Commission[5] has been used. Trips in the month of June 2016 of yellow taxis were considered. Samples of 1, 2, 4, 6, 8, and 10 million trips were extracted and processed by computing clusters with 1, 2, 4, 6, 8, 10, and 12 worker nodes separately. An evaluation of the STEAM framework has been carried out, where the insight computation included the *counts, averages, sums, minimums, and maximums* for passenger *pickups and drop offs* based on the domain-specific parameters *total amount, tip, passenger count, trip duration, and distance* on a per *hour, day, week, month, quarter, and year* basis for all neighborhoods in New York City individually. Furthermore, an origin-destination matrix has been calculated to show movements between neighborhoods. The computations are executed on *Amazon Web Services* virtual machines. EC2 instances of the type *m1.large* are used to realize an *Apache Spark* computation cluster. A dedicated *m3.medium* instance is used for the MongoDB. The computation time ideally decreases with less volume of data and more worker nodes involved in the computation. A threshold is expected to be identified at which the computation time is lesser reduced although the cluster size is further increased. This expectation complies with Amdahl's Law[19] which is originally used to estimate the gained *speedup* of multiple processor cores that solve a certain task compared to a single one solving the same task. Gustafson's Law disagrees with Amdahl's Law claiming linear scaling is possible[20]. The results of the evaluation are presented in Figure 3. The computation time increases linearly with the data quantity. Only the 1 node cluster took significantly longer to compute 10 million documents (about 15 hours) in comparison to 8 million documents (about 6 hours). This computation was repeated to exclude external irregularities, but with the same outcome. This underlines the assumption that distributed computing is required with increased data quantity in order to provide a scalable approach. The distances between the graphs decrease with increasing cluster sizes, leading to the conclusion that the speedup decreases with increased distribution. This is in conformity with Amdahl's Law and meets the aforementioned expectations.

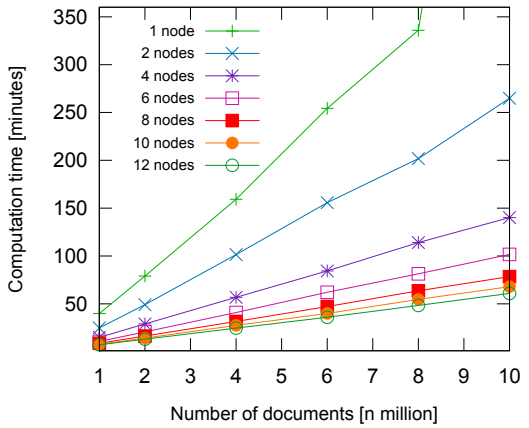


Fig. 3. The horizontal axis denotes the data quantity, the vertical axis the computation time. The upper green graph represents a cluster size of 1 computation node, the blue graph 2, the purple 4, the pink 6, the red 8, the orange 10, and the green graph at the bottom a cluster size of 12 computation nodes.

## 5. Conclusion

This paper presented STEAM, a platform for scalable spatiotemporal analytics. STEAM provides a unique way to abstract the incoming raw data from different positioning systems and adapt these to provide a generalized spatiotemporal analytics flow once the raw data is abstracted. STEAM then provides the data to external services, which are able to use the insights beyond visualization. It has been developed with scalability in mind and uses the Apache Spark framework for scalable distributed computing and the distributed database MongoDB for the provision of the analytics results. An evaluation within a testbed with different sizes of multi-machine clusters has been carried out to assess the scalability of the STEAM platform. This evaluation has proven linear scalability for different dataset sizes of a test dataset containing data from New York City taxi rides. With the ever rising size of spatiotemporal data, the increased assessment of scalable methods for spatiotemporal analytics is required in order to be able to provide insights to services in a reliable and performance-capable manner.

## Acknowledgements

This work has been carried out in the STEAM project funded by the German Federal Ministry of Education and Science (grant number 01IS12056). The authors thank B. Hanotte and D. Arbuzin for their support.

## References

1. Willems, N., van de Wetering, H., van Wijk, J.J.. Visualization of Vessel Movements. In: *Proceedings of the 11th Eurographics IEEE - VGTC Conference on Visualization*. The Eurographs Association; John Wiley & Sons, Ltd.; 2009:959–966.
2. Musa, A., Eriksson, J.. Tracking Unmodified Smartphones Using Wi-Fi Monitors. In: *Proceedings of the 10th ACM conference on embedded network sensor systems*. ACM; 2012:281–294.
3. Ester, M., Kriegel, H., Sander, J., Xu, X.. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. AAAI Press; 1996:226–231.
4. He, Y., Tan, H., Luo, W., Feng, S., Fan, J.. MR-DBSCAN: a scalable MapReduce-based DBSCAN algorithm for heavily skewed data. *Frontiers of Computer Science* 2014;8(1):83–99.
5. NYC Taxi and Limousine Commission Trip Record Data. `http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml`; 2017. Accessed: 2017-01-20.
6. Deva, B., Ruppel, P.. Location Analytics as a Service: Providing Insights for Heterogeneous Spatiotemporal Data. In: *2015 IEEE International Conference on Web Services*. 2015:353–360.
7. Hormann, K., Agathos, A.. The Point in Polygon Problem for Arbitrary Polygons. *Computational Geometry* 2001;20(3):131–144.
8. du Mouza, C., Litwin, W., Rigaux, P.. SD-Rtree: A Scalable Distributed Rtree. In: *2007 IEEE 23rd International Conference on Data Engineering*. 2007:296–305.
9. Gao, S.. Spatio-Temporal Analytics for Exploring Human Mobility Patterns and Urban Dynamics in the Mobile Age. *Spatial Cognition & Computation* 2015;15(2):86–114.
10. Ferreira, N., Poco, J., Vo, H.T., Freire, J., Silva, C.T.. Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips. *IEEE Transactions on Visualization and Computer Graphics* 2013;19(12):2149–2158.
11. Calabrese, F., Colonna, M., Lovisolo, P., Parata, D., Ratti, C.. Real-Time Urban Monitoring Using Cell Phones: A Case Study in Rome. *IEEE Transactions on Intelligent Transportation Systems* 2011;12(1):141–151.
12. Wagner, S., Willing, C., Brandt, T., Neumann, D.. Data Analytics for Location-Based Services: Enabling User-Based Relocation of Carsharing Vehicles. In: *Proceedings of the International Conference on Information Systems*. 2015:1–16.
13. Jaeger, E., Altintas, I., Zhang, J., Ludäscher, B., Pennington, D., Michener, W.. A Scientific Workflow Approach to Distributed Geospatial Data Processing Using Web Services. In: *Proceedings of the 17th International Conference on Scientific and Statistical Database Management*. Lawrence Berkeley Laboratory; 2005:87–90.
14. Mehta, P., Windolf, C., Voisard, A.. Spatio-Temporal Hotspot Computation on Apache Spark (GIS Cup). In: *24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 2016:.
15. Rahman, M.F., Suhaim, S.B., Liu, W., Thirumuruganathan, S., Zhang, N., Das, G.. ANALOC: Efficient analytics over Location Based Services. In: *2016 IEEE 32nd International Conference on Data Engineering*. 2016:1366–1369.
16. Rodriguez Garzon, S., Deva, B., Hanotte, B., Küpper, A.. CATLES: A Crowdsensing-supported Interactive World-scale Environment Simulator for Context-aware Systems. In: *Proceedings of the 2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems*. ACM; 2016:77–87.
17. Sun, H., McIntosh, S.. Big Data Mobile Services for New York City Taxi Riders and Drivers. In: *2016 IEEE International Conference on Mobile Services*. 2016:57–64.
18. Jogalekar, P., Woodside, M.. Evaluating the Scalability of Distributed Systems. In: *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*. 1998:524–531.
19. Amdahl, G.M.. Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities. In: Hill, M.D., Jouppi, N.P., Sohi, G.S., eds. *Readings in Computer Architecture*. Morgan Kaufmann Publishers Inc.; 2000:79–81.
20. Gustafson, J.L.. Reevaluating Amdahl's Law. *Commun ACM* 1988;31(5):532–533.