Design principles of a stream-based framework for mobility analysis

Loic Salmon, Cyril Ray

Noname manuscript No.
(will be inserted by the editor)

Design principles of a stream-based framework for mobility analysis

Loic Salmon · Cyril Ray

Received: date / Accepted: date

Abstract Trajectory analysis is of crucial importance in several fields as social analysis, zoology, climatology or traffic monitoring. Over the last decade, the number of mobile systems and devices recording their positions has grown significantly generating a deluge of spatial and temporal data to analyze. This increasing volume of data raises numerous issues in terms of storage, processing and extraction of information. Previous works considering movement analysis have been mainly oriented towards either archived data processing and mining or continuous handling of incoming streams. The research developed in this paper introduces the design principles of a holistic approach combining real-time processing and archived data analysis to process mobility data "on the fly". This solution aims to provide better results comparing to both purely offline and online approaches. This research considers distributed data and processing to be more efficient. The design principles are applied to maritime traffic analysis and a few representative examples are introduced to demonstrate the relevance of our approach.

Keywords Moving object database · Geostreaming · Maritime monitoring

1 Introduction

Over the past few years, the proliferation of sensors and devices recording positioning information regularly produces very large volumes of heterogeneous

Loic Salmon
Naval Academy Research Lab 29240 Brest Cedex 9, France
Tel.: +33 2 98 23 43 89
E-mail: loic.salmon@ecole-navale.fr

Cyril Ray
Naval Academy Research Lab 29240 Brest Cedex 9, France
Tel.: +33 2 98 23 36 11
E-mail: cyril.ray@ecole-navale.fr

data. This leads to many research challenges as the storage, distribution, management, processing and analysis of the large mobility data repository generated that is far from being supported by most of current spatial databases.

Mobility analysis is a generic data manipulation functionality that relates to many scientific and application domains such as traffic monitoring, environmental and ecological modeling, urban planning, sociology and robotics. Often, one of the main challenges when analyzing mobility data is to infer patterns and outliers in the datasets generated, in order to determine some novel knowledge regarding the phenomena studied. For instance, in the context of traffic monitoring, regular behaviors and abnormal trajectories can be inferred [67]. Similarly, in sociology, emerging human mobility patterns can be inferred and studied, and this according to different social communities [35]. However, most current systems do not completely provide an efficient storage and analysis of mobility data [82]. Indeed, not only moving objects continuously recorded can produce huge amounts of trajectory data, but they also need to be continuously updated.

Increasing amount of spatial and moving objects data raise some issues and challenges to store and analyze it in a good way. The problem is to determine if handling large volume and high velocities of spatial data need specific solutions to deal with spatial and mobility analysis specificities or if existing systems efficient to deal with non spatial data only need to be extended to solve this problem [8]. Some recent works have pointed out the need for a (re)definition of the problem of handling the deluge of spatial information [82] [87].

Without pretending to be exhaustive, we believe that the understanding of the following issues is crucial to build a distributed management system dedicated to spatial data storage and processing:

Data partitioning. To store and distribute in an efficient way we would like spatial and spatio-temporal data to be evenly spread over all machines of the distributed architecture. But a balanced partitioning of spatial data is more difficult because data and objects are unevenly distributed in space compared with usual data that can be randomly distributed and thus equally on all machines available. For spatial data and a fortiori moving objects we would be able to distribute them in relation to their spatial or spatio-temporal coverage area. However, the events and phenomena are not uniformly distributed spatially or temporally. This makes it difficult to evenly distribute data with a static grid, while that allocating dynamic coverage areas poses performance problems because every new record requires to move data to preserve balance on all machines.

Complexity processing. The processing of spatial and spatio-temporal is more complex. Indeed, operations on spatial data imply many space objects and entities, each of which can comprise a large number of points, lines or polygons. Moreover space operations are not only distances computing between space objects or topological operations on particular areas, but also complex analysis and spatial data mining. Mobility analysis increases the range and complexity of queries, mobile objects requiring to reconstruct each trajectory

and comparing with each other which is very expensive and generates many joins in a database.

Data Complexity. Spatial data have a complex and heterogeneous structure. Indeed, the spatial representation comprises the positions, but also lines, and polygons that can have very different shapes and sizes. This poses a problem of indexing space objects compared to non-spatial data. In addition, the time component adds complexity because if we store only the positions relative to their geographic coordinates it can lack of efficiency to process spatiotemporal queries or queries relative to movements. Many solutions exist index following certain problems [65] but considering the continuous arrival of data to be processed it's difficult to maintain an optimal structure without having to rebuild the index, which can be costly and can affect performance.

Locality and Granularity aspects. Spatial data is an abstract representation of observations. While real world has a complex infinite, we cannot extract all the information, it is often necessary to choose one data representation level for instance a trajectory can be seen as a set of points or segments. Understanding and manipulating data is therefore limited by this scale, and following spatial granularity or spatio-temporal choice, process and information that income will be irremediably affected. The inclusion of a representation at different levels of granularity is possible but generate complex process [68]. This raises also the problem of data relevance and the amount of data needed to give a good response to a query. Indeed, maybe process all the data stored in the database is not necessary, but take only a small part is enough to give an approximate answer with a better time response. Timeliness is often considered in on-line systems considering that recent data are more relevant than old data.

Spatial and temporal dependance. All the events taking place are at least partially correlated to time and space which means that a random sampling is not appropriate for spatial data processing. So it's more difficult to do statistical analysis on this data or at least necessitates to build specific algorithms taking into account spatial specificities. In the case of moving objects, for instance a solution to sample a trajectory is to keep only the records that allows to find the place where the object was located by interpolating, without losing information rather than random sampling.

Moving objects specificities. Moving objects generate a lot of data records that need to be processed regularly. That's why dealing with trajectories is handling a deluge of spatial information because both volume and velocity aspects have to be considered. Moreover, while movements are stored in a discrete way, we have to interpolate and take into consideration uncertainty of locations. Indeed, if we have to determine the actual position of a moving object, we need to consider the last record location arrived in database and infer the position thanks to last heading and speed recorded in the system.

The emergence of new systems and paradigms to deal with huge amount of data such as map-reduce [19] provide some promising solutions but still do not completely deal with moving objects [24]. This motivate a few recent works such as Map-reduce system specifically dedicated to the spatial domain [4] and particularly moving objects [52]. But still these systems are oriented towards

"a posteriori analysis" and can lack of efficiency to process data "on the fly". In most of current real-time application contexts, a successful processing system for mobility analysis have to be reactive enough and allow for anomalies detection in real-time, requiring to combine results extracted from historical data with incoming data.

Given the specificity of the analysis of mobility issues in terms of velocity, volume and low latency, we aim to design a framework for a hybrid architecture allowing tracking and analysis of moving objects in real-time. The components of this infrastructure should take into account the special nature of spatial data and manage the process of set of trajectories stored in archive or arriving on the fly. This paper gives the first principles of development of such an architecture.

The remainder of this paper is organized as follows. Section 2 provides an overview and discusses existing works oriented towards offline and online processing of mobility data before introducing hybrid processing related works concerning no spatial data. Section 3 gives a general presentation of the maritime context and the expectations of our hybrid system to handle moving objects at sea. In Section 4 we developed the principles of the online processing approach which is the focus of our hybrid-based proposal. In Section 5 we categorize the different queries that our system should be able to process by instantiating a few examples to show working and abilities of our system. Finally, Section 6 summarizes and concludes this paper.

2 Related work

Recent research works in the field of mobility analysis have been mainly oriented towards either an offline approach which stores all the data and process it on demand [75], or an online approach whose goal is to track and predict the trajectories of moving objects [60].

2.1 Offline and online approaches for handling moving objects

The mining or offline processing of historical data is characterized by a complete storage of the history of mobility data, while data is manipulated to generally retrospectively study and predict the next stages of the phenomena represented by such mobility data. In many application domains, and due to the large dataset volumes generated, response-time to any query or analysis is of crucial importance. Indeed, there is a need for some manipulation mechanisms (e.g., data structures, partitioning) to provide efficient access to the data, and in order to prevent continuous updates. Most of current works oriented to the manipulation of mobility data came from the moving object database domain [29]. Extended relational or object-oriented approaches have already integrated specialized data representation and manipulation extensions (e.g., complex data types, operators) to deal with moving objects [75],

[18]. Usual database functions can be then applied to moving objects such as data mining techniques: extraction of outliers, aggregation, clustering [36]. Such manipulation functions allow for an identification of typical behaviors or outliers [67]. Moreover, it appears that mining techniques require the distribution of data and processing when the volume of data increases considerably [53]. While offline approaches can provide valuable solutions in many cases, they do not support specific real-time functionalities and for instance how to react to some specific events.

The main principles of an online approach for trajectory analysis concern continuously tracking of objects in motion, detection and prediction of some typical behaviors as data is incoming. Such an approach is characterized by a memory-based processing where data is processed "on the fly" for better response times. Some works extend data stream management systems (DSMS) for handling spatio-temporal data and addressing the problem of real-time analysis on moving objects [60]. However this kind of approach is still constrained by memory-based processing that implies to either sample or aggregate the data using thematic, spatial or temporal criteria [38]. Another difficult aspect of the management of trajectory data is that a given analysis should be performed while the considered moving object may change its location in the upcoming stream. Meanwhile, and as some continuous queries are processed, these queries should be re-evaluated continuously as well which necessitates an incremental processing paradigm to prevent the system from complete re-evaluations [59]. Moreover, these multiple continuous queries must be executed and recomputed simultaneously while objects move. This necessitates specific approaches for sharing data manipulation and processing to bring together moving objects possibly associated to the same moving queries. Some recent works suggest some distributed processing approach performed at multiple nodes to deal with moving objects [92]. Such approaches seems more appropriate taking into account that positioning data is received from multiple different locations and that a system relying in a centralized system may be overloaded. The limitations of an online approach is that it often leads to unsatisfactory situations because some data have been deleted or altered or not income yet in the system. Indeed, even if the system is able to respond in sufficient time to a given query, accuracy is not guaranteed because of either limited amount of data considered, or alteration of the location of the moving object.

The problems and limitations mentioned above motivate our search for a hybrid solution whose objective will be to give query responses in sufficient time while maintaining appropriate accuracy and appropriateness regarding the dynamism of the system. Then let us introduce the different works related to hybrid storage and processing. To the best of our knowledge such approach haven't been used yet to deal with moving objects.

2.2 Towards a hybrid approach

The main motivations and principles of a hybrid approach have been described in [15] in which three types of queries are distinguished: those on archived data, those related to the data received in real-time, and those so-called "hybrid" queries that require to combine real-time data and query results extracted from archived data. To the best of our knowledge it was the first work to consider merging of old and recent data to process data in a database. The solution proposed by the authors of this work is to reduce the amount of archived data to process when overload occurs. The amount of archived data needed is reduced by sampling or aggregating old data to answer a hybrid query. Some others works derive from this approach, in [23] the authors try to correlate events over live and archived data streams to identify specific event while handling data that incomes massively in the system. To do this they mainly cache query results, keep new data in memory and focus on pattern correlation queries. Then they develop their own idea of events to build their complex event processing system acting in a hybrid way with financial context as an application case. Another close work [11] concerns the identification of patterns over streaming and archived data in a complex event processing context.

More recently, with the emergence of the big volume and velocity issue, a new model of architecture has been proposed. The so-called "Lambda Architecture" proposes a data management system taking into account both velocity and volume aspects with the constraint of low latency [54]. This architecture consists of three layers, a layer which corresponds to the data stored in a NOSQL database and pre-computed views related to frequently asked queries, a layer corresponding to real-time processing, and an intermediate layer that allows to merge the results of the previous two layers.

Existing hybrid systems can be classified as follows: DBMS-based systems, map-reduce-based systems and DSMS-based systems [37]. DBMS-based systems deal with high level queries (SQL-like) and encompass query planning optimization while map-reduce-based systems scale-out with fault tolerance and process huge amount of data. However, neither DBMS or map-reduce systems are real-time systems. DSMS-based systems is the only kind of system that supports real-time requirements. It enables real-time processing with a context given by the analysis of historical data, but still such systems are non easy to implement as they have to take into account the processing of large incoming real-time data with archived data. But nowadays mapreduce systems have evolved to deal with real-time requirements aspects and then can be considered as serious candidates for handling moving object processing and storage.

Indeed, at the beginning, the emergence of hadoop and mapreduce systems to deal with huge amounts of data allowed to face with high volume and process challenge. Nevertheless, this kind of system were ill-equipped to take care of velocity aspect and are not well suited for high iterative processes that is our main concern. That's why some systems have been developed to fix these

problems as Mapreduce online [17] that provide an interactive mapreduce by pipelining the map and reduce phase or others systems based on a Mapreduce incremental paradigm [66], [50] where some views are updated while data income in the system. But these systems are still batch-oriented and don't provide sufficient performances to deal with velocity and iterative processes such as machine learning algorithms.

More recently, Spark [94] had emerged as the successor of hadoop supposed to be hundred time faster than hadoop. Spark is based on RDD (Resilient Distributed Datasets) [93] which allows to keep in memory a part of the intermediate results without writing in on disk and staying fault tolerant that is really useful to perform iterative processes. Moreover some primitive operators such as filter, join or partition have been added on RDD to the mapreduce basis to partition data and process it in a DAG way [9]. With its extension Spark streaming [95], Spark allows data processing in hybrid way combining batch and streaming processing. To act as a DSMS (Data Stream Management System), Spark processes data in a "micro-batch way", considering that a stream is a mini-batch. So in spite of its well admitted efficiency, Spark stays batch-oriented and can provide poor results. For instance, if you have some period where no data is incoming in the system, old record may stay in the system without been processed while the mini-batch length has not been reached yet. Moreover, the granularity of the answers are bounded by the batch length chosen for the micro-batch.

Summingbird [13] is an illustration of the Lambda Architecture, with a batch layer a speed layer and a serving layer. The batch layer, usually works in a Map-Reduce way to process huge amounts of data whereas the speed layer process data in real-time as such system like S4 or Storm do it. The serving layer is responsible for merging the views extracted from the real-time part with the partial aggregates derived from the batch part. Such an architecture allows to write a code only once and everything will be executed a different way considering if it has to process data in online or offline way.

Apache Flink derived from the initial works on Stratosphere [5] is a distributed stream-oriented system. The systems disposes of a richer set of primitives than mapreduce, it acts as a dataflow system to perform iterative processing [27] [28]. Indeed, with delta iteration it permits to process data and chain the iterative phase in an incremental way. It includes a query optimizer that parallelizes and optimizes the workflow processing system considered even for UDF (User Defined Functions) [46] and reorder the pipelining of the operators if necessary [45]. Moreover, different kinds of windows and operators on them are available for instance triggers than are called when some events occurs, we can then imagine triggers that calls other triggers and operators when a special value incomes into the system. Finally, contrary to Spark it processes data incoming "on the fly" by pipelining directly them trough the dataflow. Following the "Flink way of thinking" batch processing is just stream processing concerning a bounded period of time. That's why Flink seems to be more stream-oriented than Spark, and thus more suitable to deal with real-time requirements.

3 Maritime application context and motivations

Maritime transportation is a domain of increasingly intense traffic (Figure 1). The monitoring and analysis of mobilities at sea is therefore crucial for safety and security reasons. For instance, these analysis of mobility and behavior should be designed to detect illegal or criminal activities, risks at sea (flow of illicit products, illegal immigration, overfishing, pollution by hazardous materials, piracy, accidents , etc.), and more generally any violation to regulations. Traffic monitoring is nowadays largely based on the continuous identification of vessel positions and trajectories and some additional functionalities such as pattern and abnormal behavior detections [67].

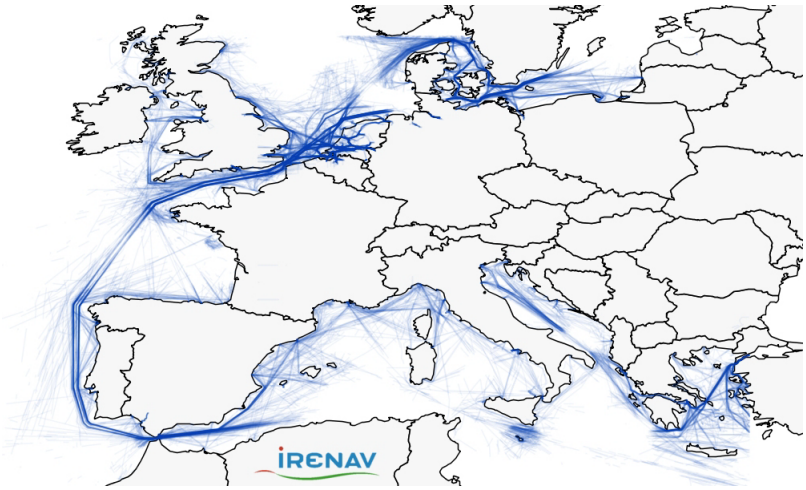

Fig. 1 Ships' trajectories, density map in Europe during one month (AIS positions, December 2010)

Practically, ships are fitted out with almost real-time position report systems whose objective is to identify and locate vessels at distance (Automatic Identification System (AIS) for example [2]). The multiplication of vessel positioning systems such as AIS but also Satellite AIS, Vessel Monitoring System (VMS) or Long Range Identification System (LRIT) contributes to the real-time availability of large traffic data at sea. The large datasets generated become difficult to manage and analyze due to the heterogeneity, large volumes and real-time components of the large datasets generated. Detecting trends and abnormal behaviors at sea still require such large-scale continuous collection of vessel positions and the development of specific spatio-temporal analysis and knowledge extraction methods [88].

Amongst the requirements, not only such a system should be able to manipulate fresh and historical data separately and jointly but also appropriate functionalities to respond to any queries in almost real-time. This clearly mo-

tivates and supports our search for a hybrid system that needs to be stream-oriented (or DSMS-based), to respond in few seconds to minutes and not hours to any queries in order to allow a maritime agent to act immediately when any problem happens. We thus propose a DSMS-based system for vessels mobility and traffic. The following section introduces the main challenges and principles retained for the design of such a system.

3.1 Goals and expectations for a monitoring maritime traffic system

In order to manage and monitor maritime traffic, it is necessary to respect some principles and requirements. Those principles raise some issues to still consider for processing of mobility data in real-time:

- *Using views for real-time analysis.* The system should give an answer within a short time period compared to a purely offline system and of better accuracy than a purely online one. However, a measure of the quality of response must be done to evaluate if the system can support execution of complicated queries and perform trajectory analysis in near real-time. To deal with real-time requirements such system will store online and offline views in main memory. The system should be able to take, translate and merge them with incoming data [33]. Views should be the cornerstone of our architecture to share processing and results and then reduce the response time.
- *An adaptive system.* This system should be reactive and adapt itself gradually as the data and queries are received to be as efficient as possible and perform incoming queries [21]. The framework must monitor the system performances and modify itself. For example, changing allocation of processes to the different nodes of the architecture to best fit with the queries already running in the system. Data that are no longer useful must be aggregated or transferred to the offline part to reduce the amount of data to process.
- *Query understanding to deal with mobility.* This system should handle a large variety of queries, this necessitates algorithms to decompose queries, analyze similarity between queries, use previous results of queries or views and merge it with incoming data. There is still a need to analyze and study the different kinds of queries related to moving objects and find similarities and common processing over them [79].
- *An autonomous and reactive processing system.* This system should handle processing and maintenance of views in accordance to queries frequently formulated by the agents. It should detect the emergence of new events and advise agents of those modifications. It might also reduce the intervention of human agents and process himself data and provide results only if necessary in a DAHP (Data Active Human Passive) way" [3].

3.2 Design of a hybrid system for moving objects

Requirements and goals of our design having been formulated in the previous section. Let us describe the principles of our hybrid architecture whose objective is to handle moving objects.

The hybrid system suggested is structured with two main components: One relates to the offline processing while the other one is responsible for the online part (cf. Figure 2). Both components can run independently in pure offline or online way storing and processing data incoming in the system, but the goal of such a system is to exploit the advantages of both approaches in order to answer hybrid queries.

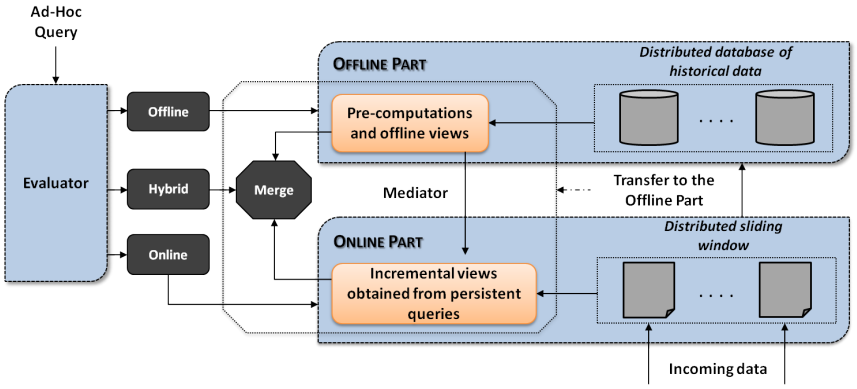

Fig. 2 Architectural principles

Online processing is performed on a distributed sliding window whose size can be changed according to the amount of data collected in real-time on the respective coverage area. Online views on continuous queries are updated and incremented while incoming data stream is processed. When a user gives query for which no synthesized summary exists, necessary data to handle the query are accessible via the sliding window. When the temporal interval of the sliding window is exceeded, the data is transferred to the historical database to perform distributed processing offline. In order to have a reactive system, summaries are performed on historical data and updated upon arrival in the database and then transferred to online part to provide real-time answers. Both online and offline parts use a distributed processing schema that still needs to be defined to take advantage of the spatial and temporal distribution of positioning records.

Furthermore, two entities have the role (cf. Figure 2) of identifying the data to extract and process, and to manage the interactions between the historical database and real-time processing system. In other words, these are the major components of our hybrid system which allows us to merge the online and

offline parts and to answer the query using the minimum data as possible in accordance to user's requirements.

The role of the *Mediator* is to manage the flows between components online and offline, preserve and store the associated views and to merge them to answer hybrid queries. The *Evaluator* analyzes the input query and tries to infer the type of request, (i.e. online, offline or hybrid) to guide, based on the identified type of query, recovery of data and information needed in our architecture. It transmits the desired data to the *Mediator* to deal with, then the *Mediator* is responsible to answer the query taking, combining or performing processing on the sliding time window or on the archive following the query type.

4 Towards a Stream-based system for moving objects

Considering our application context on maritime traffic monitoring our hybrid approach must be stream-oriented to deal with real-time requirements. Let us examine the challenges and difficulties involved in the design of a DSMS-based approach and those directly inherited from the DSMS part.

4.1 Challenges involved in a distributed data stream management system for moving objects

In a DSMS, a query consists of stream operators organized in a directed acyclic graph (DAG) or workflow [9]. Query operators are connected via queues and each stream operator has an in-memory state composed by the tuples needed to perform the operation whose is responsible for. The DAG uses a pipeline paradigm where each result produced by an operator is transmitted to the following operator(s) in the workflow. All data are processed in main memory and some tuples or results cached are shared between the different queries which run simultaneously.

Handling moving object in this context raises some additional issues that should be taken into account to design a distributed moving object stream management system. Let us sum up the challenges associated to the design of a distributed DSMS-based system for mobility analysis.

- *Distribution and optimization of the queries.* This system can be modeled as a dataflow where incoming data flows and queries should be re-evaluated accordingly [21]. Therefore, such system should create and delete novel processes when necessary to stay efficient as the system evolves. Online systems must also process multiple queries simultaneously, then incoming data associated to different queries must be brought and processed together in order to reduce number of operations. Finally this system must re-order the operators that process data and choose the best efficient query execution plan.

- *Evaluation and approximation.* An important problem to deal with in on-line systems is that data are processed in memory only. Therefore, the amount of data manipulated in memory should be reduced when processing the re-sampling, compressing load-shedding... [38]. Another approach is to design specific algorithms that give approximate solutions by a data evaluation performed at regular times. Data summaries or data views can be also used as a sort of incremental processing paradigm [33].
- *Distributed schema processing.* The two previous challenges must be extended in the context of a distributed system. Such distribution brings common additional issues such as fault tolerance, scalability and data distribution. In the context of our approach view placement, dynamic allocation of the operators, data transfer from one node to another must be also taken into account [80]. This implies to distribute both data and processing along the nodes of our architecture. Distribution can be done according to spatial coverage, temporal extent or by taking into account moving object movement patterns [83].
- *A hybrid stream-based approach.* In the context of hybrid processing stream-oriented, the online part must be the cornerstone of this system that can be combined with offline results. Therefore, this system must be able to merge query results of online and offline parts as well as making a difference between the online and offline parts of a hybrid query.

To the best of our knowledge, the first research work to address the integration of the fields of data stream management and moving objects is introduced in [71]. In this paper, the author defines a data model for handling moving object as data streams but seems ill-equipped to deal with some cases (only moving points are considering and not moving regions). The system developed provides some sampling algorithms to manipulate moving objects in order to reduce the amount of data stored in memory [76], using sliding windows at different levels of granularity [69] or using summaries [77]. However, and despite the fact that the amount of data processed in memory is relatively limited, multiple queries and integration of on-line and archived data is not taken into account. The only aspects concerning distribution and optimization of queries are inherited from TelegraphCQ [14], but the problem is that evaluation and approximation techniques have not been implemented into an only one system or applied to extend TelegraphCQ.

Other related work focuses on scalability issues where the author introduces a shared-execution paradigm [16], in other words a join between data and queries to deal with coordinated execution of multiple queries applied to several objects in a moving object context [60]. Another focus is made on the incremental query processing [59] to avoid complete re-evaluation of persistent queries into the system. Indeed, to avoid a complete reevaluation of a given query, difference is made positive and negative updates when previous results or summaries of data are used via predicate windows [32]. In [90] the author address the problem into account incremental processing and multiple queries handling but some views are still stored on disk. In SOLE [58] the same au-

Table 1 State of the art Resume for Stream Moving Objects Systems

Research works	Origin system	Management and query repartition	Semantic & Evaluation	Distribution
PLACE [60]	Nile [41] (online) Predator (offline)	-Scheduling [40] -Pipeline -Multi-join [39]	-Predicate windows [32] -Incremental evaluation [34] [91] -Views handling [33] -Spatio-temporal histogram [25]	Place * QTP model [90]
StreamSecondo [96]	Secondo (offline) [18] Possible link with Parallel-Secondo [52]		-Stream Algebra [43] -Windows implementation [44] -Stream types	
Kostas Patroumpas works	TelegraphCQ (online) [14] Possible link for offline with Hermes(PostGis) [74]	-Sharing paradigm (Psoup) [16] -Adaptive processing (eddy)[10] -Dynamic scheduling [86]	-ST Sampling [76] -Windows aggregation [73] -Windows at different granularity level [72] -Index at different granularity level [77]	Flux [81]
SCUBA [62]	CAPE [78]	-Plan migration [97] -Adaptive scheduling [85] -Cluster sharing paradigm [62]	-Clustersheddy [63] -Aggregation	D-CAPE [84]
GeoInsight [48]	Microsoft StreamInsight [7]	-Fusing horizontal & vertical [7] -Stream partitionning knn range queries [56]	-Event-based [7] -Views derived from archive in-memory [48] -Native support for ST stream [6]	
Infosphere Streams ITS [12]	SPADE [7]	-scheduling component [89] -operator fusing [49] -basic PE (Processing Element)	-map-matching [6] -shortest path	-Datafaflow [12]
Zaghreb laboratory works	-TelegraphCQ -Implementation in java	-General framework for MO [30]	-Uncertainty handling -Trajectory buffering [55]	

thors proposes to deal with moving objects in a online way only by processing data in-memory, but this work doesn't take care of distribution aspects.

The work introduced in SCUBA uses datamining techniques such as clustering to reduce the amount of data and time processing, using a shared-cluster based execution paradigm and load-shedding and applied to moving objects [62]. Despite its interest, a limitation of this work is that although it is appropriate when dealing with moving objects with relatively predictable behavior in some constrained urban networks, this being not the case in the maritime domain.

StreamSecondo [96] extends Secondo [18] by providing an algebra to deal with spatial streams. However, the management and query repartition aspects are not the main concern of this work which focuses on spatial objects rather than on moving objects. Some windows and spatial operators have been implemented but don't seem to be sufficient to our application context.

In [30] the authors suggest a general framework to deal with moving objects knowing about the limits of TelegraphCQ in [71] and inspired by the algebra defined in [96]. It can be used as a basis for developing a geospatial DSMS but doesn't take into account management and query repartition aspect.

In [48] the authors address the design of a system for mobility data processing extending the following CEP (Complex Event Processing system) [7]. The authors propose to merge old data derived as sketches with incoming data to handle moving objects. Moreover, spatio-temporal streams and spatial queries like knn-query and range query have been developed. However, the system seems designed for networks and the distribution aspect is not addressed.

IBM Infosphere streams ITS [12] provides a modular distributed framework to process positioning data. It deals with management and query repartition aspect by reordering [89] and fusing operators [49]. Although spatial data and operators are not implemented and the main concern is rather on map-matching or shortest path finding than mobility analysis.

Another works have emerged with proliferation of cloud-streaming solutions as Storm and S4 [64] by extended them to deal with mobility analysis [92], [31]. However, these systems distribute the process but do not appear to deal with query optimization and approximations issues.

MOCEP (Complex Event Processing system for Moving Object) (cf. section 5) is the system that we propose to deal with mobility analysis in an outdoor context. It considers movement in term of events to process mobility data. A hybrid approach combining real-time processing and archived data analysis is suggested to handle moving objects "on the fly". Distribution over several nodes is also considered to improve data processing. It requires a more precise definition of events necessary to express and store views into the system.

Table 2 Pro and Cons of State of the art Stream Moving Objects Systems

Research works	Proposition	Advantages	Drawbacks
PLACE SEA-CNN SINA	-Stream processing of moving objects -Incremental processing -Multiple concurrent queries handling	-sharing paradigm -incremental evaluation -predicate windows	-Views stored on disk -Unavailable -Not distributed
StreamSecondo	-Algebra for real-time spatial processing	-Definition of spatial stream objects -Definition of spatial and windows operators	-Basic query sharing -Basic operator ordering -Need for more operators -Not distributed
Kostas Patroumpas works	-Sampling -Windows at different granularity levels -Specific synopses	-Granularity handling -Amount of data in memory reduce -Quick time response	-Limited sharing & ordering abilities -Only moving point are handled -Not distributed -Works not implemented in only one system
SCUBA	-Shared cluster paradigm -Load shedding and compression	-Better sharing than PLACE -Less data and in memory process	-Network based -Relevant for MO that moves together -Accuracy of the response (aggregation & deletion)
IBM Infosphere	ITS	-Distributed -Dataflows processing -Windows handling	-Basic use case -Network based a priori -Incremental processing ? -Spatial types ?
GeoInsight	Coupling archived & real-time data	-Views of archived data in main memory (sketch) -Spatial operators	-Network-based a priori -Incremental processing ? -Only few spatial queries (knn, range)
Zaghreb laboratory works	General framework for MO Algrebra for stream MO	-Uncertainty handling -Trajectory buffering	-Preliminary works -Not implemented on only one system
MOCEP Complex Event Processing system for Moving Objects	General distributed framework for MO	- Sharing paradigm system -Incremental processing - In-memory views - Distributed in a DAG way -Specific windows and triggers -Query plan optimization	-Not implemented yet -Better definition of views needed -Event modeling still preliminar

4.2 Stream processing principles to handle moving objects

Considering both challenges for handling moving objects in a DSMS-based approach 4.1 and principles of our hybrid system 3.2 to deal with operational context, let us propose the following architecture (cf. Figure 3) which is currently under development and falls into the conception of a stream-oriented hybrid system.

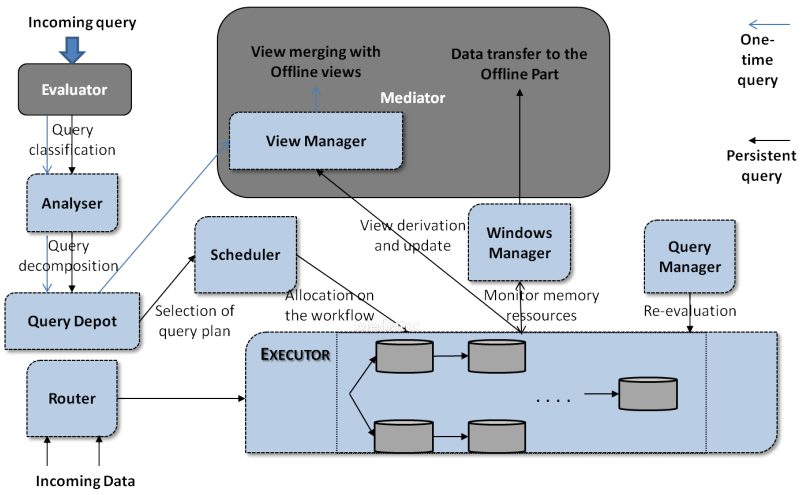

Fig. 3 Online architectural principles

There are two different inputs, one related to the system which collects the data and processes it as it incomes, and one related to the user which formulates and adds his query to the pool of persistent queries or wants an answer to a hybrid query. The link between these two components is made by the *Executor* which is the distributed data flows processing engine whose role is to join moving objects with queries in order to share processing and data along the different operators.

Incoming streams are received, pre-processed and supplied to the stream operators by the *Router* while new queries are analyzed and translated in the form of workflow (or query plan) by the *Analyzer*. As a new query is specified in the system, there are two different options following the result given by the *Evaluator* on the hybrid part. When the incoming query is persistent, then it is added to the system to run continuously whereas if it is a one-time query it will be executed once and processed mainly using views. The *Analyzer* extracts the online part related to the query by desegregating the query into online and offline sub-queries. The query plans associated to the online queries are then stored in a repository (*Query Depot*) and managed by the *Query Manager* that stores, indexes and ensures the periodic re-execution of continuous queries. The

Executor, or dataflow based engine gives the link between the data sent by the *Router* and the queries which have been translated into scheduling processes and optimized by the *Scheduler* according to queries already running in the *Executor*. Finally, the *Memory Manager* is responsible for the management of the sliding windows and the transfer of data from the online part to offline one via the *Mediator* in case of obsolescence of data, while the *View Manager* is in charge of view maintenance obtained from successive executions of continuous queries.

When the incoming query is executed once then the system checks if the online sub-query(or a part of it) is already running in the pool of queries (in the *Query Depot*). If this is the case the system takes the view associated to the query via the *View Manager*, if not the system computes the result using the distributed sliding window and generates the associated view as it's done for a persistent query that run continuously.

5 MOCEP : a Complex Event Processing system for Moving Object

5.1 Different kinds of queries for moving objects

Since development of moving object databases (MOD), a lot of different works have been made to process and a particularly one query such as continuous knn-query [91] or range query [47]. Instead, we aim to provide a system that will be able to deal with most of the different queries that can be formulated by an agent in a general framework as in [57]. The different kind of moving objects queries can be classified into three different kinds of queries [20]. First of all, queries can concern comparison of trajectories with specific locations or POI (Point Of Interest). An example of a so-called P-query could be "What are the moving objects that have been close to this specific point?". Secondly, in some queries called R-queries trajectories and regions can be involved, for instance we can be interested in finding trajectories that cross a specific spatiotemporal region or finding regions that are usually crossed by moving objects. An illustration of such queries could be "What are the regions that are the most crossed between the period time T ?". Finally, T-queries are queries related to similarity finding in a set of trajectories. This can be relevant to identify some meeting, collision phenomena between moving objects but also to identify specific patterns. An example of such a query could be "What are the moving objects that could intersect the trajectory of this specific moving object during the next five minutes?". Those are the different queries that our system should be able to deal with. We can also categorize queries considering the time period concerned by the query as it is done in [65] for index. Past query will concern old data, whereas now-queries will refer to current and recent data. Predictive queries need recent and old data following the accuracy and the time prediction of the query. Indeed, if we want to estimate the future location of moving objects in 2 minutes, this position can be computed with

precedent heading and velocity with a tiny error, while estimate the position in 10 minutes can require recent data heading and velocity to compare with old data to have a better accuracy. Finally, in our hybrid stream-oriented system we can classify queries considering if they are to be executed once in the system so we'll talk about one-time queries or if they have to run continuously as data incomes in the system so we'll talk about continuous queries.

5.2 Event paradigm to deal with mobility analysis

We can model displacement of moving objects as events. The notion of events to deal with trajectory have been proposed recently in [70] where a Complex Event Recognition system is used to identify some typical behaviors in the maritime traffic domain. The notion of event and hybrid pattern matching (over stream and archived data) developed in [11] can be extended in our moving object context. Let's consider the following assumptions :

Definition 1 : Basic Event. *A basic event for moving object is noted as follows E=E(EventId,TypeId,time,<a_1,...,a_n>), where EventId identifies the moving object event, Typeid refers to the kind of moving object event considered, the time value defines the moment where the event took place and <a_1,...,a_n> corresponds to the attributes specific to the event. Here we'll consider only one kind of events that is the belonging of a moving object to a spatio-temporal area at a determinate time. For the following part of this paper a Basic Event will refer to this specific event.*

Definition 2 : Complex Event. *A complex event for moving object is defined as E=E(EventId, C=<e_1 Opr e_2 Opr... e_n>, ts, te, <a_1,...,a_n>) where C is a set of basic or complex events linked with each other by Opr event constructor that can be disjunction, conjunction and so on while ts is the time corresponding to the beginning of the event and te the time of end. For this paper, complex events will be limited only to conjunction of basic event previously defined. Then complex moving object event associated to a moving object will correspond only to the spatio-temporal areas crossed by this moving object.*

These event definitions that still need to be extended, allow us to define results from queries execution by patterns. These patterns expressed as a sequence of events can provide some views at another level to filter and compare elements with incoming streams into the system. In order to do that, we'll have to correlate the on-line execution of queries defined in the previous section 3.2 with specific patterns recorded as views in-memory or on disk considering the free space in memory. Here also we can distinguish three kind of generic patterns [51]. Individual patterns focus on the behavior of a specific moving object in order to identify some specific patterns that occurs periodically for the moving object considered. Pairwise patterns concerns process on pairs of moving objects to detect some collision or avoidance phenomena. Finally, aggregate

Table 3 Representative table of different queries that the system should deal with

Kind of queries	Past queries	Present queries	Predictive queries
P-query	"What are the moving objects that have been closed from this specific point?"	"What are the specific points closest to this moving object?"	"What are the moving object that will be close to this point in the next five minutes?" "What will be the location of this moving object in t time?"
R-query	"What are the moving objects that have crossed this area during the period T ?" "What are the region that are more crossed? "	"What are the moving objects inside of the area ? " "What are the moving object that are leaving the area ? "	"Which moving objects will enter in this area ? "
T-query	"What is a representative trajectory from this kind of moving object?"	"What is the mean speed of the moving objects in this specific area? "	"What are the moving objects that are supposed to intersect during the next five minutes?"

patterns are extracted to find specific behaviors over multiples trajectories like moving clusters and derived (flocks, convoys, swarms ...), frequent trajectory patterns or overcrowded areas. Those patterns have to evolve in an incremental way for the online part to reduce the process due to complete reevaluation.

5.3 General statements and preliminary elements of the system

Let us introduce the operational elements of our system. The system we have chosen to extend is the Apache Flink system described on section 2.2, because it seems the more suitable to deal with real-time requirements allowing to make hybrid processing. The proposed solution takes also into account elements and design principles previously defined (cf. Figure 2 and Figure 3)

Let us describe different elements and the way they are handled in the Apache Flink system. A *Window Assigner* is responsible for the assignment of elements to one or more windows, in a similar way to our *WindowsManager* defined in section 4.2. Each window owns a *Trigger* that forces the evaluation and execution of on a window or a part of it. The *Trigger* is called when an element is inserted into the windows or some time events occurs (typically when no activity have been recorded for a certain time period). A window can be evaluated several times depending on the nature of the *Trigger* and data incoming into the system. The *Evictor* is an optional complement responsible for deleting some data that are to old and can be involved independently or after the *Trigger* action.
Flink deals with iteration in a specific way. For a classical iterative algorithm as machine learning algorithms, the entire input is consumed into an operator chain to produce the next version of the partial solution, and the partial solution is used to fed the following step while stop conditions have not been reached (convergence criterion or maximum number of iterations). There is also the notion of delta iteration that can be very useful in our context. Rather than fully recompute all data at each step, just new data are evaluated and merged with results derived from the previous iteration.

Concerning the distribution, data are split on the different nodes accordingly to their spatial location. Basically, we'll mesh space by using grid cells of regular size. We know about the skew phenomena and some future works and refinement will be done to study the best repartition. The advantages of splitting data and queries accordingly to their spatial area is that if each moving object doesn't move from its area, we'll have less data to examine to retrieve corresponding data and compute results. The problems occurs when the moving object leaves its area, so we'll have to transfer the moving queries and views associated to this moving object to the node responsible for the new covering area in a similar way to the QTP model proposed in [90].

5.4 Illustrative queries for maritime traffic monitoring

In our system, queries can be persistent or one-time considering if they are continuously running into the system of they are executed once. Persistent queries are the cornerstone of our system because they produce views that are continuously updated and synthesize some interesting results. These synopses can be used directly to answer other queries. Here we present representative persistent queries relevant for maritime traffic monitoring:

5.4.1 Illustrative persistent queries

Persistent queries related to trajectory reconstruction. For each trajectory, while a new location is incoming in the system the new location is added to the trajectory. When the record is made on a different node comparing to node where last record have been registered, then views and moving queries transfer is executed. The view corresponding to the trajectory reconstruction is the set of the different locations reported during the last L minutes, where L is the size of the sliding window chosen that need to be determined that could be variable. Views idea necesitates some mechanisms to refresh and have a view related to the time period considered. To update the view for trajectory construction, updates are done by appending new records as explained and old records are deleted via *Evictors*.

Persistent queries related to next location. With AIS system, each record is done following the speed of the ship. We can thus determine the next location that will be registered in the system and raise some triggers or alerts if no position are emitted from the boats or if the location doesn't fit with the previous that has income into the system. To estimate the next position of a vessel we can take into account the ship kind, the previous heading and velocity without forgetting uncertainty aspects. We can then in a similar way to [76] deleting unnecessary data that are not relevant to describe ship's movement.

Persistent queries related to a trajectory pattern. Here we wants the system to resume as synopses the representative path that vessels generally use to cross some areas like in [26]. For instance if we are interested by the typical road followed by cargo ships in the Atlantic Ocean it can be relevant to aggregate trajectories of all boats that have crossed the area to extract the different representative patterns and behaviors. We could use some relevant datamining techniques dedicated to trajectory mining as Dbscan or clustering techniques to determine spatio-temporal patterns as in [22]. Here we propose to use the Fpgrowth algorithm developed in [42] whose objective is to find frequent patterns in transaction and time serie databases. This mining technique seems suitable and fit with the event model proposed in the previous section 5.2. Indeed, we can mesh space with tiles that will be from different size from the area size for every node of the system and use Fpgrowth to find trajectory patterns in a similar way to [61]. A basic event is the fact that a

moving object has recorded its location in a tile and represents an item while a complex event is a sequence of basic events constituting the whole trajectory of the moving object and corresponds to a transaction. So we'll generate a tree of frequent patterns (frequent trajectories/subtrajectories) from the transaction database (set of trajectories).

Some other queries are formulated by the agents as they observe suspicious behavior. Our system takes advantage of the views derived from persistent queries to answer to these one-time queries. Let us give some representative examples:

5.4.2 Illustrative one-time queries

One-time Query: "Is this vessel following a 'normal' trajectory?" The role of maritime agents is to monitor maritime traffic in order to prevent some accident from happening or reacts quickly when some problems occurs. In this context, where the agent have to look after maybe until several hundred of ships at a time, a query that automatically detects some strange behavior can be useful. To determine the "normality" of a trajectory, we want to compare actual trajectories with trajectory pattern recorded in the system as views as in [22]. We can then use the benefits of the views related to trajectory pattern

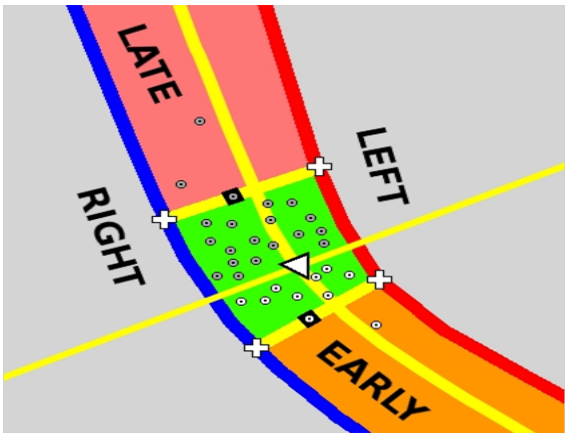

Fig. 4 Spatio-temporal corridor to detect abnormal trajectories and behaviors [22].

extraction and compare with actual trajectories recorded in the system in the views concerning trajectories reconstruction by using a Jaccard distance. If Jaccard distance between the specific vessel and the trajectory patterns registered is high, we can analyze this boat displacement over the last few days to see if there is some periodic pattern or if its movement seems normal.

Here, we can see the relevance of patterns derived from events for our system, because it allows to filter elements and focus on suspicious behaviors.

One-time Query: "What are the fishing areas near from Brest?" Identify fishing areas is relevant because it allows to make some difference between legal and illegal fishing to preserve environment and fish species diversity. In order to do that, we restrain the study to fishing boats, behaviors of vessels currently fishing is specific and can be identify easily. The boat moves at a low speed and makes some loops to fish, then for each fishing vessel we compute the coverage area of fishing and intersect with coverage areas of other fishing boats to extract a density map of activity and obtain fishing areas as proposed in [1]. Those fishing areas extracted from the offline part and incremented in

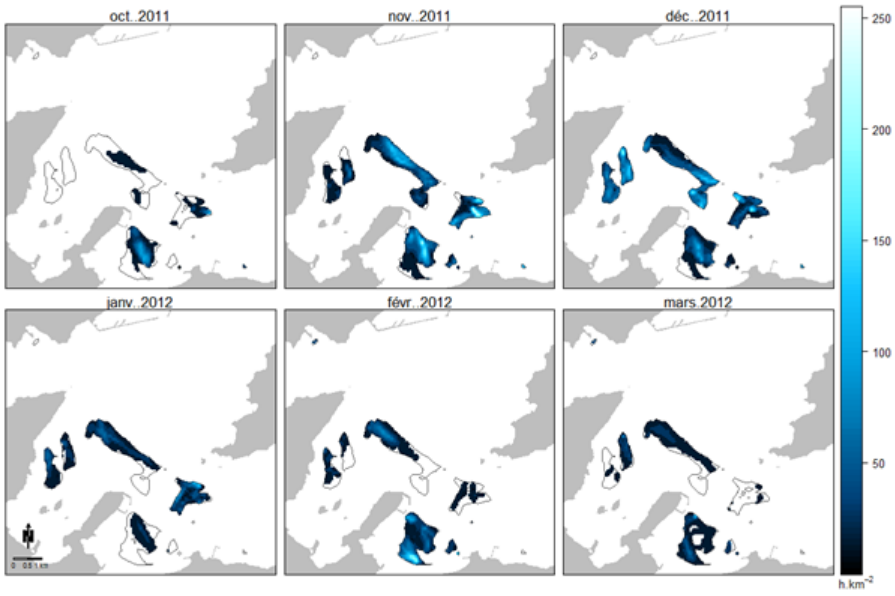

Fig. 5 Estimated fishing activity for scalop dredging vessels per month in 2011-2012 (expressed in fishing hours/km)[1].

real-time are then compared with fishing boat currently in movement and clearly identify fisherman in fraud. The problem here, is that some fishermen shut down their system of position reporting when they are in illegal areas. To extend the query, a good idea could be to identify some "gaps" in fishing boat trajectories to infer which one belong to malicious anglers.

5.5 Discussion

The research developed in this paper introduced the principles of a hybrid approach for mobility mining combining real-time analysis with information extracted from archived data. The system developed is mainly oriented to the online part of the architecture in order to deal with real-time requirements. We expect that most queries will be processed using only the online part and views associated to the analysis of archived data. Such system should be autonomous and process data and advise agents when necessary. The advantages of such an approach is its likely fast response-time and its reactivity to new events in a operational context, but can lack of efficiency for stream mining or running of complex queries.The few examples presented here give some statements in the mechanisms involved but some of them need to be studied and improved. For instance, the idea of event must be investigated and should allow to answer queries of this kind "What are the moving objects that leaves this area in a high speed with a heading change?". An another thing to investigate is the distribution of the data and the design of algorithms that should take advantage from this distribution in an incremental way. Finally, views need to be enriched and defined at different granularity levels to answer the queries and to limit computes in our system.

6 Conclusion

Over the past few years the emergence and proliferation of mobile and sensor-based systems have generated a significant increase of spatial and temporal data in terms of volume and update frequency. The maritime domain in particular had recently to face with an explosion of positioning data that requires a reevaluation of existing methods and systems to deal with maritime traffic monitoring. Previous works related to trajectory analysis have been directed towards either mining archived historical data (offline) or continuous processing of incoming data streams (online). In this work we have introduced the design principles of a hybrid approach combining both online and offline approaches to process maritime traffic data. The hybrid architecture suggested is stream-based and deal with real-time requirements of operational contexts enriched by the analysis of archived data. It has been instantiated in a few examples to illustrate the mechanisms and algorithms used. Ongoing work concerns the development and implementation of the event processing paradigm proposed and the evaluation of our use case examples.

References

1.
2. *Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band.* Recommendation ITU-R M.1371-5 (02/2014), 2014.

3. D. J. Abadi, D. Carney, U. Çetintemel, M. Cherniack, C. Convey, C. Erwin, E. F. Galvez, M. Hatoun, A. Maskey, A. Rasin, A. Singer, M. Stonebraker, N. Tatbul, Y. Xing, R. Yan, and S. B. Zdonik. Aurora: A data stream management system. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, California, USA, June 9-12, 2003*, page 666, 2003.
4. A. Aji, F. Wang, H. Vo, R. Lee, Q. Liu, X. Zhang, and J. Saltz. Hadoop gis: A high performance spatial data warehousing system over mapreduce. *Proc. VLDB Endow.*, 6(11):1009–1020, Aug. 2013.
5. A. Alexandrov, R. Bergmann, S. Ewen, J. Freytag, F. Hueske, A. Heise, O. Kao, M. Leich, U. Leser, V. Markl, F. Naumann, M. Peters, A. Rheinländer, M. J. Sax, S. Schelter, M. Höger, K. Tzoumas, and D. Warneke. The stratosphere platform for big data analytics. *VLDB J.*, 23(6):939–964, 2014.
6. M. H. Ali, B. Chandramouli, B. S. Raman, and E. Katibah. Spatio-temporal stream processing in microsoft streaminsight. *IEEE Data Eng. Bull.*, 33(2):69–74, 2010.
7. M. H. Ali, C. Gerea, B. S. Raman, B. Sezgin, T. Tarnavski, T. Verona, P. Wang, P. Zabback, A. Kirilov, A. Ananthanarayan, M. Lu, A. Raizman, R. Krishnan, R. Schindlauer, T. Grabs, S. Bjeletich, B. Chandramouli, J. Goldstein, S. Bhat, Y. Li, V. D. Nicola, X. Wang, D. Maier, I. Santos, O. Nano, and S. Grell. Microsoft CEP server and online behavioral targeting. *PVLDB*, 2(2):1558–1561, 2009.
8. L. Anselin. What is special about spatial data? alternative perspectives on spatial data analysis. pages 63–77, 1989.
9. A. Arasu, B. Babcock, S. Babu, J. Cieslewicz, M. Datar, K. Ito, R. Motwani, U. Srivastava, and J. Widom. Stream: The stanford data stream management system. Technical Report 2004-20, Stanford InfoLab, 2004.
10. R. Avnur and J. M. Hellerstein. Eddies: Continuously adaptive query processing. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA.*, pages 261–272, 2000.
11. M. Balazinska, Y. Kwon, N. Kuchta, and D. Lee. Moirae: History-enhanced monitoring. In *CIDR 2007, Third Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 7-10, 2007, Online Proceedings*, pages 375–386, 2007.
12. A. Biem, E. Bouillet, H. Feng, A. Ranganathan, A. Riabov, O. Verscheure, H. N. Koutsopoulos, and C. Moran. IBM infosphere streams for scalable, real-time, intelligent transportation services. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, Indianapolis, Indiana, USA, June 6-10, 2010*, pages 1093–1104, 2010.
13. P. O. Boykin, S. Ritchie, I. O'Connell, and J. Lin. Summingbird: A framework for integrating batch and online mapreduce computations. *PVLDB*, 7(13):1441–1451, 2014.
14. S. Chandrasekaran, O. Cooper, A. Deshpande, M. J. Franklin, J. M. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, F. Reiss, and M. A. Shah. Telegraphcq: Continuous dataflow processing. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, California, USA, June 9-12, 2003*, page 668, 2003.
15. S. Chandrasekaran and M. Franklin. Remembrance of streams past: Overload-sensitive management of archived streams. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases*, VLDB '04, pages 348–359, 2004.
16. S. Chandrasekaran and M. J. Franklin. Psoup: a system for streaming queries over streaming data. *VLDB J.*, 12(2):140–156, 2003.
17. T. Condie, N. Conway, P. Alvaro, J. M. Hellerstein, K. Elmeleegy, and R. Sears. Mapreduce online. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, pages 21–21, Berkeley, CA, USA, 2010. USENIX Association.
18. V. T. de Almeida, R. H. Guting, and T. Behr. Querying moving objects in secondo. In *Proceedings of the 7th International Conference on Mobile Data Management*, MDM '06, pages 47–52. IEEE Computer Society, 2006.
19. J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, OSDI'04, pages 10–10, Berkeley, CA, USA, 2004. USENIX Association.

20. K. Deng, K. Xie, K. Zheng, and X. Zhou. Trajectory indexing and retrieval. In *Computing with Spatial Trajectories*, pages 35–60. 2011.
21. A. Deshpande, Z. Ives, and V. Raman. Adaptive query processing. *Found. Trends databases*, 1(1):1–140, Jan. 2007.
22. T. Devogele, L. Etienne, and C. Ray. Maritime monitoring. In *Mobility Data: Modeling, Management, and Understanding*, pages 221–239. 2013.
23. N. Dindar, B. G, P. Lau, A. zal, M. Soner, and N. Tatbul. Dejavu: declarative pattern matching over live and archived streams of events. In U. etintemel, S. B. Zdonik, D. Kossmann, and N. Tatbul, editors, *SIGMOD Conference*, pages 1023–1026. ACM, 2009.
24. A. Eldawy and M. F. Mokbel. The era of big spatial data. In *31st IEEE International Conference on Data Engineering Workshops, ICDE Workshops 2015, Seoul, South Korea, April 13-17, 2015*, pages 42–49, 2015.
25. H. G. Elmongui, M. F. Mokbel, and W. G. Aref. Spatio-temporal histograms. In *Advances in Spatial and Temporal Databases, 9th International Symposium, SSTD 2005, Angra dos Reis, Brazil, August 22-24, 2005, Proceedings*, pages 19–36, 2005.
26. L. Etienne, T. Devogele, and A. Bouju. Spatio-temporal trajectory analysis of mobile objects following the same itinerary. *Advances in Geo-Spatial Information Science*, 10:47–57, 2012.
27. S. Ewen, S. Schelter, K. Tzoumas, D. Warneke, and V. Markl. Iterative parallel data processing with stratosphere: an inside look. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*, pages 1053–1056, 2013.
28. S. Ewen, K. Tzoumas, M. Kaufmann, and V. Markl. Spinning fast iterative data flows. *CoRR*, abs/1208.0088, 2012.
29. L. Forlizzi, R. H. Güting, E. Nardelli, and M. Schneider. A data model and data structures for moving objects databases. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pages 319–330, New York, NY, USA, 2000. ACM.
30. Z. Galic, M. Baranovic, K. Krizanovic, and E. Meskovic. Geospatial data streams: Formal framework and implementation. *Data Knowl. Eng.*, 91:1–16, 2014.
31. A. Garz, A. A. Benczr, C. I. Sidl, D. Tahara, and E. F. Wyatt. Real-time streaming mobility analytics. In X. Hu, T. Y. Lin, V. Raghavan, B. W. Wah, R. A. Baeza-Yates, G. Fox, C. Shahabi, M. Smith, Q. Y. 0001, R. Ghani, W. Fan, R. Lempel, and R. Nambiar, editors, *BigData Conference*, pages 697–702. IEEE, 2013.
32. T. M. Ghanem, W. G. Aref, and A. K. Elmagarmid. Exploiting predicate-window semantics over data streams. *SIGMOD Record*, 35(1):3–8, 2006.
33. T. M. Ghanem, A. K. Elmagarmid, P. Larson, and W. G. Aref. Supporting views in data stream management systems. *ACM Trans. Database Syst.*, 35(1), 2010.
34. T. M. Ghanem, M. A. Hammad, M. F. Mokbel, W. G. Aref, and A. K. Elmagarmid. Incremental evaluation of sliding-window queries over data streams. *IEEE Trans. Knowl. Data Eng.*, 19(1):57–72, 2007.
35. F. Giannotti, M. Nanni, D. Pedreschi, F. Pinelli, C. Renso, S. Rinzivillo, and R. Trasarti. Unveiling the complexity of human mobility by querying and mining massive trajectory data. *The VLDB Journal*, 20(5):695–719, Oct. 2011.
36. F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 330–339. ACM, 2007.
37. L. Golab and T. Johnson. Data stream warehousing. In *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*, pages 1290–1293, 2014.
38. L. Golab and M. T. Özsu. Issues in data stream management. *SIGMOD Rec.*, pages 5–14, 2003.
39. M. A. Hammad, W. G. Aref, and A. K. Elmagarmid. Query processing of multi-way stream window joins. *VLDB J.*, 17(3):469–488, 2008.
40. M. A. Hammad, M. J. Franklin, W. G. Aref, and A. K. Elmagarmid. Scheduling for shared window joins over data streams. In *VLDB*, pages 297–308, 2003.

41. M. A. Hammad, M. F. Mokbel, M. H. Ali, W. G. Aref, A. C. Catlin, A. K. Elmagarmid, M. Y. Eltabakh, M. G. Elfeky, T. M. Ghanem, R. Gwadera, I. F. Ilyas, M. S. Marzouk, and X. Xiong. Nile: A query processing engine for data streams. In *Proceedings of the 20th International Conference on Data Engineering, ICDE 2004, 30 March - 2 April 2004, Boston, MA, USA*, page 851, 2004.
42. J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA.*, pages 1–12, 2000.
43. Y. Huang and C. Zhang. New data types and operations to support geo-streams. In *Geographic Information Science, 5th International Conference, GIScience 2008, Park City, UT, USA, September 23-26, 2008. Proceedings*, pages 106–118, 2008.
44. Y. Huang and C. Zhang. Interval-based nearest neighbor queries over sliding windows from trajectory data. In *MDM 2009, Tenth International Conference on Mobile Data Management, Taipei, Taiwan, 18-20 May 2009*, pages 212–221, 2009.
45. F. Hueske, A. Krettek, and K. Tzoumas. Enabling operator reordering in data flow programs through static code analysis. *CoRR*, abs/1301.4200, 2013.
46. F. Hueske, M. Peters, M. Sax, A. Rheinländer, R. Bergmann, A. Krettek, and K. Tzoumas. Opening the black boxes in data flow optimization. *CoRR*, abs/1208.0087, 2012.
47. D. V. Kalashnikov, S. Prabhakar, S. E. Hambrusch, and W. G. Aref. Efficient evaluation of continuous range queries on moving objects. In *Database and Expert Systems Applications, 13th International Conference, DEXA 2002, Aix-en-Provence, France, September 2-6, 2002, Proceedings*, pages 731–740, 2002.
48. S. J. Kazemitabar, U. Demiryurek, M. H. Ali, A. Akdogan, and C. Shahabi. Geospatial stream query processing using microsoft SQL server streaminsight. *PVLDB*, 3(2):1537–1540, 2010.
49. R. Khandekar, K. Hildrum, S. Parekh, D. Rajan, J. L. Wolf, K. Wu, H. Andrade, and B. Gedik. COLA: optimizing stream processing applications via graph partitioning. In *Middleware 2009, ACM/IFIP/USENIX, 10th International Middleware Conference, Urbana, IL, USA, November 30 - December 4, 2009. Proceedings*, pages 308–327, 2009.
50. W. Lam, L. Liu, S. Prasad, A. Rajaraman, Z. Vacheri, and A. Doan. Muppet: Mapreduce-style processing of fast data. *Proc. VLDB Endow.*, 5(12):1814–1825, Aug. 2012.
51. Z. Li. Spatiotemporal pattern mining: Algorithms and applications. In *Frequent Pattern Mining*, pages 283–306. 2014.
52. J. Lu and R. H. Güting. Parallel SECONDO: practical and efficient mobility data processing in the cloud. In *Proceedings of the 2013 IEEE International Conference on Big Data, 6-9 October 2013, Santa Clara, CA, USA*, pages 17–25, 2013.
53. Q. Ma, B. Yang, W. Qian, and A. Zhou. Query processing of massive trajectory data based on mapreduce. In *Proceedings of the First International CIKM Workshop on Cloud Data Management, CloudDb 2009, Hong Kong, China, November 2, 2009*, pages 9–16, 2009.
54. N. Marz. *Big data : principles and best practices of scalable realtime data systems*. O'Reilly Media, [S.l.], 2013.
55. E. Meskovic, D. Osmanovic, Z. Galic, and M. Baranovic. Generating spatio-temporal streaming trajectories. In *37th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2014, Opatija, Croatia, May 26-30, 2014*, pages 1130–1135, 2014.
56. J. Miller, M. Raymond, J. Archer, S. Adem, L. Hansel, S. Konda, M. Luti, Y. Zhao, A. Teredesai, and M. H. Ali. An extensibility approach for spatio-temporal stream processing using microsoft streaminsight. In *Advances in Spatial and Temporal Databases - 12th International Symposium, SSTD 2011, Minneapolis, MN, USA, August 24-26, 2011, Proceedings*, pages 496–501, 2011.
57. M. F. Mokbel and W. G. Aref. Gpac: Generic and progressive processing of mobile queries over mobile data. In *Proceedings of the 6th International Conference on Mobile Data Management*, MDM '05, pages 155–163, New York, NY, USA, 2005. ACM.
58. M. F. Mokbel and W. G. Aref. SOLE: scalable on-line execution of continuous queries on spatio-temporal data streams. *VLDB J.*, 17(5):971–995, 2008.

59. M. F. Mokbel, X. Xiong, and W. G. Aref. SINA: scalable incremental processing of continuous queries in spatio-temporal databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, France, June 13-18, 2004*, pages 623–634, 2004.
60. M. F. Mokbel, X. Xiong, M. A. Hammad, and W. G. Aref. Continuous query processing of spatio-temporal data streams in place. *Geoinformatica*, pages 343–365, 2005.
61. M. Morzy. Mining frequent trajectories of moving objects for location prediction. In *Machine Learning and Data Mining in Pattern Recognition, 5th International Conference, MLDM 2007, Leipzig, Germany, July 18-20, 2007, Proceedings*, pages 667–680, 2007.
62. R. V. Nehme and E. A. Rundensteiner. SCUBA: scalable cluster-based algorithm for evaluating continuous spatio-temporal queries on moving objects. In *Advances in Database Technology - EDBT 2006, 10th International Conference on Extending Database Technology, Munich, Germany, March 26-31, 2006, Proceedings*, pages 1001–1019, 2006.
63. R. V. Nehme and E. A. Rundensteiner. *ClusterSheddy* : Load shedding using moving clusters over spatio-temporal data streams. In *Advances in Databases: Concepts, Systems and Applications, 12th International Conference on Database Systems for Advanced Applications, DASFAA 2007, Bangkok, Thailand, April 9-12, 2007, Proceedings*, pages 637–651, 2007.
64. L. Neumeyer, B. Robbins, A. Nair, and A. Kesari. S4: Distributed stream computing platform. In *Proceedings of the 2010 IEEE International Conference on Data Mining Workshops*, ICDMW '10, pages 170–177. IEEE Computer Society, 2010.
65. L.-V. Nguyen-Dinh, W. G. Aref, and M. F. Mokbel. Spatio-temporal access methods: Part 2 (2003 - 2010). *IEEE Data Eng. Bull.*, 33(2):46–55, 2010.
66. C. Olston, G. Chiou, L. Chitnis, F. Liu, Y. Han, M. Larsson, A. Neumann, V. B. Rao, V. Sankarasubramanian, S. Seth, C. Tian, T. ZiCornell, and X. Wang. Nova: Continuous pig/hadoop workflows. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, pages 1081–1090, New York, NY, USA, 2011. ACM.
67. G. Pallotta, M. Vespe, and K. Bryan. Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction. *Entropy*, 15(6):2218–2245, 2013.
68. K. Patroumpas. Multi-scale window specification over streaming trajectories. *J. Spatial Information Science*, 7(1):45–75, 2013.
69. K. Patroumpas. Multi-scale window specification over streaming trajectories. *J. Spatial Information Science*, pages 45–75, 2013.
70. K. Patroumpas, A. Artikis, N. Katzouris, M. Vodas, Y. Theodoridis, and N. Pelekis. Event recognition for maritime surveillance. In *Proceedings of the 18th International Conference on Extending Database Technology, EDBT 2015, Brussels, Belgium, March 23-27, 2015.*, pages 629–640, 2015.
71. K. Patroumpas and T. K. Sellis. Managing trajectories of moving objects as data streams. In *Spatio-Temporal Database Management, 2nd International Workshop STDBM'04, Toronto, Canada, August 30, 2004*, pages 41–48, 2004.
72. K. Patroumpas and T. K. Sellis. Multi-granular time-based sliding windows over data streams. In *TIME 2010 - 17th International Symposium on Temporal Representation and Reasoning, Paris, France, 6-8 September 2010*, pages 146–153, 2010.
73. K. Patroumpas and T. K. Sellis. Subsuming multiple sliding windows for shared stream computation. In *Advances in Databases and Information Systems - 15th International Conference, ADBIS 2011, Vienna, Austria, September 20-23, 2011. Proceedings*, pages 56–69, 2011.
74. N. Pelekis, E. Frentzos, N. Giatrakos, and Y. Theodoridis. Hermes: Aggregative lbs via a trajectory db engine. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1255–1258, New York, NY, USA, 2008. ACM.
75. N. Pelekis, Y. Theodoridis, S. Vosinakis, and T. Panayiotopoulos. Hermes - a framework for location-based data management. In *In Proceedings of EDBT*, pages 1130–1134, 2006.

76. M. Potamias, K. Patroumpas, and T. K. Sellis. Sampling trajectory streams with spatiotemporal criteria. In *18th International Conference on Scientific and Statistical Database Management, SSDBM 2006, 3-5 July 2006, Vienna, Austria, Proceedings*, pages 275–284, 2006.
77. M. Potamias, K. Patroumpas, and T. K. Sellis. Online amnesic summarization of streaming locations. In *Advances in Spatial and Temporal Databases, 10th International Symposium, SSTD 2007, Boston, MA, USA, July 16-18, 2007, Proceedings*, pages 148–166, 2007.
78. E. A. Rundensteiner, L. Ding, T. M. Sutherland, Y. Zhu, B. Pielech, and N. K. Mehta. CAPE: continuous query engine with heterogeneous-grained adaptivity. In *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*, pages 1353–1356, 2004.
79. M. A. Sakr and R. H. Güting. Group spatiotemporal pattern queries. *GeoInformatica*, 18(4):699–746, 2014.
80. M. A. Shah, J. M. Hellerstein, and E. A. Brewer. Highly-available, fault-tolerant, parallel dataflows. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, France, June 13-18, 2004*, pages 827–838, 2004.
81. M. A. Shah, J. M. Hellerstein, S. Chandrasekaran, and M. J. Franklin. Flux: An adaptive partitioning operator for continuous query systems. In *Proceedings of the 19th International Conference on Data Engineering, March 5-8, 2003, Bangalore, India*, pages 25–36, 2003.
82. S. Shekhar, V. Gunturi, M. R. Evans, and K. Yang. Spatial big-data challenges intersecting mobility and cloud computing. In *Proceedings of the Eleventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*, MobiDE '12, pages 1–6, New York, NY, USA, 2012. ACM.
83. X. Sun, A. Yaagoub, G. Trajcevski, P. Scheuermann, H. Chen, and A. Kachhwaha. P^2est: parallelization philosophies for evaluating spatio-temporal queries. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data, BigSpatial@SIGSPATIAL 2013, Nov 4th, 2013, Orlando, FL, USA*, pages 47–54, 2013.
84. T. M. Sutherland, B. Liu, M. Jbantova, and E. A. Rundensteiner. D-CAPE: distributed and self-tuned continuous query processing. In *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November 5, 2005*, pages 217–218, 2005.
85. T. M. Sutherland, Y. Zhu, L. Ding, and E. A. Rundensteiner. An adaptive multi-objective scheduling selection framework for continuous query processing. In *Ninth International Database Engineering and Applications Symposium (IDEAS 2005), 25-27 July 2005, Montreal, Canada*, pages 445–454, 2005.
86. T. Urhan and M. J. Franklin. Dynamic pipeline scheduling for improving interactive query performance. In *VLDB 2001, Proceedings of 27th International Conference on Very Large Data Bases, September 11-14, 2001, Roma, Italy*, pages 501–510, 2001.
87. R. R. Vatsavai, A. Ganguly, V. Chandola, A. Stefanidis, S. Klasky, and S. Shekhar. Spatiotemporal data mining in the era of big spatial data: Algorithms and applications. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*, BigSpatial '12, pages 1–10, New York, NY, USA, 2012. ACM.
88. M. Vodas, N. Pelekis, Y. Theodoridis, C. Ray, V. Karkaletsis, S. Petridis, and A. Miliou. Efficient ais data processing for environmentally safe shipping. *SPOUDAI Journal of Economics and Business*, 63(3-4):181–190, 2013.
89. J. L. Wolf, N. Bansal, K. Hildrum, S. Parekh, D. Rajan, R. Wagle, K. Wu, and L. Fleischer. SODA: an optimizing scheduler for large-scale stream-based distributed computer systems. In *Middleware 2008, ACM/IFIP/USENIX 9th International Middleware Conference, Leuven, Belgium, December 1-5, 2008, Proceedings*, pages 306–325, 2008.
90. X. Xiong, H. G. Elmongui, X. Chai, and W. G. Aref. Place: A distributed spatio-temporal data stream management system for moving objects. In *8th International Conference on Mobile Data Management (MDM 2007), Mannheim, Germany, May 7-11, 2007*, pages 44–51, 2007.
91. X. Xiong, M. F. Mokbel, and W. G. Aref. SEA-CNN: scalable processing of continuous k-nearest neighbor queries in spatio-temporal databases. In *Proceedings of the 21st*

International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan, pages 643–654, 2005.
92. Z. Yu, Y. Liu, X. Yu, and K. Q. Pu. Scalable distributed processing of K nearest neighbor queries over moving objects. *IEEE Trans. Knowl. Data Eng.*, 27(5):1383–1396, 2015.
93. 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 *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, NSDI'12, pages 2–2, Berkeley, CA, USA, 2012. USENIX Association.
94. M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: Cluster computing with working sets. In *2nd USENIX Workshop on Hot Topics in Cloud Computing, HotCloud'10, Boston, MA, USA, June 22, 2010*, 2010.
95. M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, and I. Stoica. Discretized streams: Fault-tolerant streaming computation at scale. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, SOSP '13, pages 423–438, New York, NY, USA, 2013. ACM.
96. C. Zhang, Y. Huang, and T. Griffin. Querying geospatial data streams in SECONDO. In *17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2009, November 4-6, 2009, Seattle, Washington, USA, Proceedings*, pages 544–545, 2009.
97. Y. Zhu, E. A. Rundensteiner, and G. T. Heineman. Dynamic plan migration for continuous queries over data streams. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, France, June 13-18, 2004*, pages 431–442, 2004.