様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科研費 KAKENHI

# 科学研究費助成事業　　研究成果報告書

令和　５ 年　 ６ 月 １３ 日現在

機関番号：８２６２６

研究種目：若手研究

研究期間：2020～2022

課題番号：２０Ｋ１９８０６

研究課題名（和文）GeoFlink: A real-time and highly scalable processing framework for the spatial data streams

研究課題名（英文）GeoFlink: A real-time and highly scalable processing framework for the spatial data streams

研究代表者

ＳＨＡＩＫＨ　ＳＡＬＭＡＮ（SHAIKH, SALMAN AHMED)

国立研究開発法人産業技術総合研究所・情報・人間工学領域・主任研究員

研究者番号：３０７４２６２１

交付決定額（研究期間全体）：（直接経費）　　3,100,000円

研究成果の概要（和文）：このプロジェクトは、スケーラブルでリアルタイムの空間データ ストリーム管理システム GeoFlink の研究と開発に焦点を当てています。
GeoFlink は、空間データ タイプ、インデックス、空間データ ストリームに対する連続クエリをサポートします。 クエリを効率的に処理するために、グリッドベースのインデックスが導入されています。 GeoFlink は、さまざまなジオメトリでの空間範囲、kNN、および結合クエリをサポートします。
このプロジェクトでは、3 つの会議論文と 1 つの雑誌論文を出版しました。 GeoFlink はオープンソースであり、産総研の IP として登録されています。

研究成果の学術的意義や社会的意義
Our proposed framework GeoFlink enables low-latency continuous queries (range, knn, and join) processing over spatial data streams. The GeoFlink, being real-time spatial data processing framework, can be used for target marketing, disaster management, autonomous driving, robots path guidance, etc.

研究成果の概要（英文）：With the advancement in data collection technologies, there is an increase in spatial data. Spatial data is huge and many time requires real-time processing. This project focuses on the research and development of a scalable and real-time spatial data stream management system GeoFlink.
GeoFlink extends Apache Flink to support spatial data types, indexes and continuous queries over spatial data streams. To enable efficient processing of continuous queries and for the effective data distribution across computing cluster nodes, a gird-based index is introduced. GeoFlink supports spatial range, spatial kNN and spatial join queries on point, multi-point, line, multi-line, polygon and multi-polygon geometry types. Extensive experimental study on real spatial data streams proves that GeoFlink achieves significantly higher query throughput than ordinary Flink processing.
In this project, we published 3 conference and 1 journal papers. GeoFlink is open source and is registered as an AIST IP.

研究分野：Data Streams and Distributed Computing

キーワード：GeoFlink　Scalable Processing　Spatial Stream　Continuous Queries　Spatial Indexing　Range Query　Knn Query　Join Query

## １．研究開始当初の背景

**Spatial data** or more specifically 2D spatial data refers to the geographical data obtained from devices integrated with GPS sensors, including smart phones, car navigation systems, smart watches, kids/animals tracking belts, etc. All the 2D spatial data consists of at least two attributes, i.e., x and y geo-coordinates (also known as longitudes and latitudes). Besides the traditional 2D spatial points, spatial data includes lines, multi-lines, polygons and multi-polygons. Recently there is a tremendous increase in the spatial data generation due to the increase in the use of devices with GPS sensors. With the increasing availability of the spatial data, the demand for its processing and analysis is also increasing.

**Need for Spatial Data Processing:** Spatial data processing and analysis have several applications in different domains including marketing, disaster management, autonomous driving, military services, robotics, kids/patients/animals tracking, medical science, etc., to name a few. In many cases, spatial data is huge and require real-time processing to avoid heavy financial and human loss. For instance, having been able to access the precise locations of all the humans in a disaster affected city via their smart phones, we would like to guide them to the nearest safe location in a real-time manner to avoid possible life loss. For a big city like Tokyo, this include continuous and real-time processing of millions of tuples/second related to over 38 million people trajectories [1]. The existing spatial data processing frameworks like ESRI [4] and PostGIS [5] are mainly designed for static data and not for spatial stream processing and are not scalable to handle such a huge data in real-time. Such a large-scale real-time computation is only possible with the help of state-of-the-art distributed and horizontally scalable big data processing platforms like Apache Spark , Apache Flink , Apache Kafka , etc. However, none of these platforms is designed for spatial data processing and hence cannot process spatial queries efficiently. These big data processing platforms treat spatial data as ordinary text data and hence randomly distribute incoming data to available nodes without considering the spatial proximity of the data, hence resulting in increased querying cost. Beside these, there exist some open-source libraries and platforms for the processing of spatial data on these scalable bigdata platforms, for instance, a) GeoSpark [2]: a library for the spatial data processing on Apache Spark, b) ST-Hadoop [3]: a Hadoop-based framework for the processing　of very large spatial data available in Hadoop Distributed File System, however both of them can only handle static data and does not support real-time query processing.

Hence, to process the huge data streams in real-time, we need the power of big data platforms which are highly distributed and horizontally scalable. However, to get the spatial data streams processed by these platforms, spatial data structures, data types and operators need to be implemented which is quite challenging due to the data-driven programming models of these platforms.

**Data-driven Programming:** It is a programming model where the data itself controls the flow of the program and not the program logic. Big data platforms like Apache Spark and Apache Flink uses this style of programming, by applying a series of transformations on the data, to support horizontal scalability.

The goal of this project is to propose data types, data structures and algorithms for the real-time processing of huge and dynamic spatial data streams by utilizing existing big data platforms. To avoid reinventing the wheel and to take the advantage of the existing state-of-the-art big data stream processing engines, we propose to use the Apache Flink as the base engine.

**Key Scientific Questions:**
This research will address the following key scientific questions.
1. How to represent the spatial data objects in the big data platforms for efficient processing?
2. How to structure the dynamic spatial data streams, such that the data maintains spatial proximity when distributed among the big data platform's cluster nodes?
3. Which data structures must be used to obtain a throughput of millions of tuples/second in the presence of highly dynamic data stream (i.e., with continuous insertions and deletions)?

4. What are the different useful spatial data operators?

２．研究の目的

The purpose of this project is to be able to process highly dynamic spatial data streams efficiently by extending one of the state-of-the-art big data streaming platforms as the base system, which in our case is Apache Flink. This requires proposing spatial data types, efficient and scalable spatial data structures and effective algorithms for the uniform distribution of the dynamic data streams.

Why Apache Flink? At present, Apache Spark and Apache Flink are the two main big-data platforms, both of which claim to support stream processing besides the traditional batch processing. To enable stream processing, Apache Spark relies on micro-batching while Apache Flink does not rely on such micro-batching and processes the incoming stream tuples as soon as they arrive. In contrast to Flink streaming model, micro-batching in Spark results in small processing delay as it groups the incoming streaming tuples to achieve higher throughput. Since the focus of this work is the real-time processing, Flink is a natural choice to be used as the base framework.

Scientific significance, and originality of the research project:
At the moment, there exist only one active and commonly used spatial data processing framework, i.e., GeoSpark [2], which is based on the Apache Spark platform. It is a library for the static spatial data processing using Apache Spark's batch processing. Since it cannot support real-time processing, it is not suitable for the processing and analysis of continuous spatial data streams requiring real-time response. To support efficient query evaluation, GeoSpark define spatial index structures on the spatial data attributes, i.e., on the x and y coordinates (or longitude and latitude), including Grid, R-Tree and Quad-Tree [2]. The index is generated on the static datasets, which may be used for the efficient query processing, however if there is any change in the dataset, the index needs to be regenerated as in the data-flow programming model, we cannot modify part of the index. This indexing does not only support query processing but also helps in the uniform data distribution of the spatial data by keeping in view the spatial proximity to reduce the data shuffling among the cluster nodes. However, in case of data streams, we cannot take the advantage of the similar data indexing and data partitioning as the data arrives continuously and we have no prior information of the data distribution. Besides, index creation is a costly operation and cannot serve the purpose in case of dynamic data streams.

Hence, we need some data structures, which do not need to be physically updated as the new data arrives, or the old data expires. Instead, based on the spatial attributes, the incoming data must be hashed as it arrives to the appropriate cluster node(s) to preserve the spatial data proximity. For instance, if we have four cluster nodes, hence we would like to divide the incoming stream tuples in four groups, such that the tuples in each group preserve the spatial proximity. Each group is then hashed to one cluster node, to minimize the data shuffling during query processing and hence can improve the overall throughput. For this sake, logical grid-indexing could be useful, as most of the spatial data queries involve the computation of nearest neighbors within certain distance of the queried point, and grid-indexing preserves the spatial proximity for the 2D spatial data.

３．研究の方法

This work will elucidate the following to address the key scientific questions above:

1. Since the big-data platforms including the Apache Flink, does not natively support spatial data types or objects, we will define a spatial data layer for Flink, so that it can support different spatial data types including Point, Lines, Polygons, etc.

2. The next and the most important challenge is to hash (logical index) the incoming spatial data stream in such a way that the data maintains spatial proximity when distributed among the big data cluster nodes. This could be achieved by utilizing the grid structure, like the one shown in Fig. 1, where each grid cell has length *l*. Each incoming stream tuple is assigned a unique grid cell id based on its location coordinates and the tuples belonging to the near-by cells are hashed to the same cluster node. Given the query point *q*, and cell length *l*, we can easily identify the number of spatial objects that lie within certain distance $d$ of $q$ without computing the distance between $q$ and the dataset points. For instance, in Fig. 1, assuming that blue circle denotes the boundary of $d$ from *q*, then the objects in cells within red boundary or layer $L_1$ of cell containing $q$ are guaranteed to be within distance $d$ of *q*. Layer $L_1$(q) can be defined as follows:

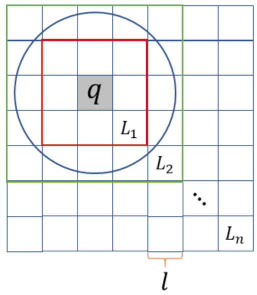


*Figure 1: Grid Structure*

$$L_1(q) = \{C_{u,v} | u = x \pm 1, v = y \pm 1, C_{u,v} \neq C_{x,y}\}$$

Similarly, we can identify the points which lie greater than distance $d$ of $q$ (cells outside green boundary), hence the distance computation needs to be done only for the cells in green boundary and excluding the red boundary. This could be helpful in the efficient nearest neighbors' search queries which is the basis of majority of spatial queries.

3. The nearest neighbor search is more challenging in case of spatial objects of type line and polygon, where an object may lie in more than one cell grid or even in more than one cluster node. Hence, we plan to extend the idea presented above for these spatial objects and devise algorithms for it.
4. Another important contribution of this work is the identification and definition of different spatial operators. Since the operations like overlap, intersect, etc. are specific to spatial objects, hence they need to be defined in Apache Flink following the data-driven programming which is another challenge.

## ４．研究成果

The main outcome of this project is GeoFlink (Figure 2 shows GeoFlink Architecture), which is a scalable and distributed framework for the real-time spatial data stream management and processing.

GeoFlink extends Apache Flink to support spatial data types, indexes and continuous queries over spatial data streams. To enable efficient processing of continuous queries and for the effective data distribution across computing cluster nodes, a gird-based index is introduced. GeoFlink supports spatial range, spatial kNN and spatial join queries on point, multi-point, line, multi-line, polygon and multi-polygon geometry types. Extensive experimental study on real spatial data streams proves that GeoFlink achieves significantly higher query throughput than ordinary Flink processing.

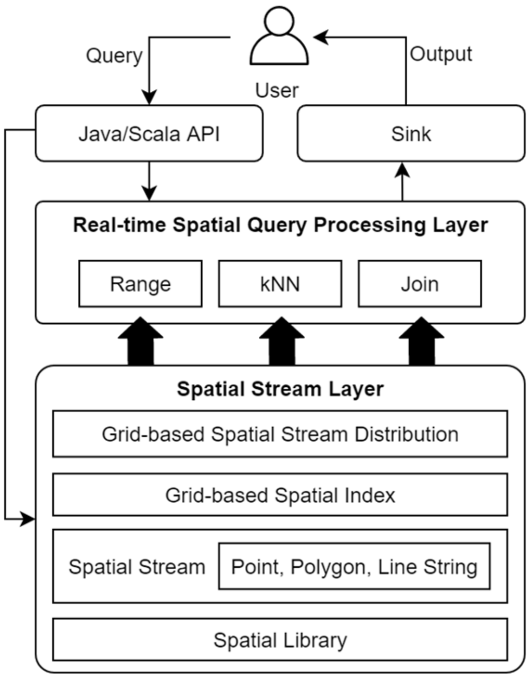


*Figure 2 GeoFlink Architecture*

In this project, we could publish 3 conference [6][7][9] and 1 journal [8] papers. The complete framework is available on Github as an open source [10] and is registered as an AIST intellectual property.

References:


[1] United Nations, "The World's Cities in 2016", March 12, 2017.

[2] Jia Yu, Zongsi Zhang, Mohamed Sarwat, "Spatial Data Management in Apache Spark: The GeoSpark Perspective and Beyond", Geoinformatica Journal, 23. 10.1007/s10707-018-0330-9, 2018.

[3] Louai AlarabiEmail authorMohamed F. MokbelMashaal Musleh, “ST-Hadoop: a MapReduce framework for spatio-temporal data”, Geoinformatica Journal, Volume 22, Issue 4, pp 785–813, 2018

[4] ESRI, https://www.esri.com/en-us/home, accessed October 23, 2019

[5] PostGIS, https://postgis.net/, accessed October 23, 2019

[6] Salman Ahmed Shaikh, Hiroyuki Kitagawa, Akiyoshi Matono, and Kyoung-Sook Kim, A Framework for Real-time and Scalable Trajectory Stream Processing and Analysis, ACM SIGSPATIAL 2022, Seattle, Washington.

[7] Masaya Yamada, Hiroyuki Kitagawa, Salman Ahmed Shaikh, Toshiyuki Amagasa, and Akiyoshi Matono, Streaming Augmented Lineage: Traceability of Complex Stream Data Analysis, iiWAS2022.

[8] Salman Ahmed Shaikh, Hiroyuki Kitagawa, Akiyoshi Matono, Komal Mariam and Kyoung-sook Kim, GeoFlink: An Efficient and Scalable Spatial Data Stream Management System, in IEEE Access, vol. 10, pp. 24909-24935, 2022, doi: 10.1109/ACCESS.2022.3154063.

[9] Salman Ahmed Shaikh, Hiroyuki Kitagawa, Akiyoshi Matono, and Kyoung-Sook Kim, A Framework for Real-time and Scalable Trajectory Stream Processing and Analysis, ACM SIGSPATIAL 2022, Seattle, Washington

[10] https://github.com/aistairc/SpatialFlink

５．主な発表論文等

〔雑誌論文〕　計1件（うち査読付論文　1件／うち国際共著　1件／うちオープンアクセス　1件）

| | |
|---|---|
| １．著者名<br>Salman Ahmed Shaikh, Hiroyuki Kitagawa, Akiyoshi Matono, Komal Mariam, Kyoung-Sook Kim | ４．巻<br>10 |
| ２．論文標題<br>GeoFlink: An Efficient and Scalable Spatial Data Stream Management System | ５．発行年<br>2022年 |
| ３．雑誌名<br>IEEE Access | ６．最初と最後の頁<br>24909-24935 |
| 掲載論文のDOI（デジタルオブジェクト識別子）<br>10.1109/ACCESS.2022.3154063 | 査読の有無<br>有 |
| オープンアクセス<br>オープンアクセスとしている（また、その予定である） | 国際共著<br>該当する |

〔学会発表〕　計3件（うち招待講演　0件／うち国際学会　3件）

| |
|---|
| １．発表者名<br>Salman Ahmed Shaikh, Komal Mariam, Hiroyuki Kitagawa and Kyoung-Sook Kim |
| ２．発表標題<br>GeoFlink: A Distributed and Scalable Framework for the Real-time Processing of Spatial Streams |
| ３．学会等名<br>CIKM '20 Proceedings of the 29th ACM International Conference on Information & Knowledge Management（国際学会） |
| ４．発表年<br>2020年 |

| |
|---|
| １．発表者名<br>Masaya Yamada, Hiroyuki Kitagawa, Salman Ahmed Shaikh, Toshiyuki Amagasa, and Akiyoshi Matono |
| ２．発表標題<br>Streaming Augmented Lineage: Traceability of Complex Stream Data Analysis |
| ３．学会等名<br>iiWAS2022（国際学会） |
| ４．発表年<br>2022年 |

| |
|---|
| １．発表者名<br>Salman Ahmed Shaikh, Hiroyuki Kitagawa, Akiyoshi Matono, and Kyoung-Sook Kim |
| ２．発表標題<br>A Framework for Real-time and Scalable Trajectory Stream Processing and Analysis |
| ３．学会等名<br>ACM SIGSPATIAL 2022（国際学会） |
| ４．発表年<br>2022年 |

〔図書〕　計0件

〔産業財産権〕

〔その他〕

－

6．研究組織

| | 氏名<br>（ローマ字氏名）<br>（研究者番号） | 所属研究機関・部局・職<br>（機関番号） | 備考 |
|---|---|---|---|

7．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

8．本研究に関連して実施した国際共同研究の実施状況

| 共同研究相手国 | 相手方研究機関 |
|---|---|