JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Spatiotemporal Big Data Processing Systems: A Comprehensive Review

K. Venkateswara Rao
Department of Computer Science and Engineering,
CVR College of Engineering, Hyderabad, Telangana State, India
kv.rao@cvr.ac.in

Abstract: Motivation: Spatiotemporal data deals with spatial and temporal aspects of the data. Big data has been gaining a strong focus of global interest and continuously attracting the attention of academia, industry, government and other organizations. Context: Continuous arrival of spatiotemporal data in varying formats from different domains has been increasing its volume and posing challenges in storing, managing, accessing and processing. Distributed data processing systems are appropriate for this kind of spatiotemporal big data because such systems architectures are scalable. Objective: To conduct literature survey, discuss and compare state of the art Spatiotemporal Big Data processing systems with respect to underlying platform and various features supported. Methods: The methodology involves identifying digital libraries and web resources, collecting state of the art research papers in the field of Spatiotemporal Big Data Processing Systems, selecting the relevant papers, eliminating duplicates from various sources, extracting the information from selected papers, performing analysis, organizing and presenting the views as results. Results: The state of the art systems for Spatiotemporal Big Data Processing are categorized, compared and discussed based on data types supported, partitioning, indexing and queries supported. Novelty: This survey outlines future research directions such as integrating Artificial Intelligence, Machine Learning, and Deep Learning models to spatiotemporal big data infrastructures for discovering hidden knowledge.

***Index Terms* – Spatiotemporal, Big data, Spatiotemporal Information Systems, Spatiotemporal Data Processing**

I. INTRODUCTION

Availability of mobile computing and location based services enabled the collection of spatial and temporal data. Spatiotemporal data has many application domains such as ecology, transportation, medicine, forestry, biology, geophysics, agriculture, oceanography, environment and meteorology. Managing Space and time, massive size and complexity associated with spatiotemporal data created many challenges for efficient processing by the applications. With the rise of spatiotemporal big data and its applications in various domains, there is a need for horizontally scalable distributed systems to store, manage, and process spatiotemporal big data. Researchers from both industry and academia have been working in collaboration to fulfil these requirements. The spatiotemporal big data processing systems that have been developed in the last few years are mostly based on MapReduce framework of Apache Hadoop, Spark and NoSQL [1] databases. Most of these systems have been built by adding a layer on top of existing spatial systems or extending the core of existing systems to support spatiotemporal data processing. A full equipped spatiotemporal big data management system is required to serve requirements of wide range of spatiotemporal applications. The requirements include Storing, Indexing [2, 3], Querying, Analysis [4], Visualization [5], Mining [6], Machine Learning with Neural Networks [7] and Deep Learning [8, 9] of spatiotemporal big data. Review of literature on current state is always pivotal to advance the state of the art. Hence this comprehensive review on the whole ecosystem of spatiotemporal big data processing systems is conducted.

Even though existing spatial systems such as SpatialHadoop [10], GeoSpark [11] are efficient for spatial operations, they have performance issues when adopted them for processing spatiotemporal queries. Adopting spatial systems for executing spatiotemporal queries such as range query will suffer from the following: (1) The spatial indices are not well suited to support spatiotemporal queries because the indices are geared for supporting spatial queries that result in scanning through irrelevant data to answer the spatiotemporal query. (2) The system internal is not aware of the spatiotemporal aspects of the data objects. One possible way to recognize nature of spatiotemporal data is by adding another dimension to spatial index.

The research community has been developing spatiotemporal big data processing systems either from scratch or by extending big data processing platforms such as Hadoop, Spark, NoSQL, and Python libraries. Hence this review on spatiotemporal big data systems categorize them based on the development criteria as Hadoop based, Spark based, NoSQL based and others.

This paper is organized as follows. Section two outlines research methodology used to review state of the art literature, Section three elaborates Hadoop-based Spatiotemporal Big Data Infrastructures, Section four discusses Spark-based Spatiotemporal big data platforms, NoSQL-based Spatiotemporal big data systems are described in section five. Other Spatiotemporal Big Data Infrastructures are described in section six. Section seven presents comparison of all the systems reviewed in this article. Conclusion and Future Research Directions are presented in section eight.

III. RESEARCH METHODOLOGY

The purpose of this systematic review paper is to collect and identify state of the art research papers in the field of Spatiotemporal Big Data Processing Systems. The methodology used in the literature review consists of planning, implementing, and investigation of results. The first stage contains formulation of the review, identifying the requirements and rules including a) research questions, b) paper extraction, c) and selection of relevant papers for the review. Well known web resources and digital libraries have been identified and searched with the keywords related to spatiotemporal big data systems for extracting the relevant research papers till date. Older papers that are less relevant are rejected and also filtered duplicated papers if any from different sources. The results of the first stage are presented in Table 1.

Table 1. Digital Libraries and Web sources and Number of Papers collected

Name	Digital Library	Web Source	Number of papers considered
IEEE Digital Library	√		7
ACM Library	√		6
Science Direct (Elsevier)	√		2
Springer	√		3
Google Scholar		√	5
DOAJ Open Access		√	7
Research Gate		√	5

The second stage is aimed to extract the relevant information from the identified papers mentioned as references in this article. The last stage presents comprehensive review of different distributed systems for processing Spatiotemporal Big Data. Following sections are the result of this systematics review process.

IIII. HADOOP BASED SPATIOTEMPORAL BIG DATA SYSTEMS

3.1 ST-Hadoop

ST-Hadoop [12] is an open source software system that offers a support for spatiotemporal big data. It injects spatiotemporal data awareness inside Hadoop software system. It is an extension to the Hadoop and SpatialHadoop. ST-Hadoop cluster consists of one master node and a set of slave nodes. The master node breaks a MapReduce job into a set of smaller tasks and instructs slave node to execute the task. ST-Hadoop contains language, Indexing, MapReduce, and operations layers in its design.

ST-Hadoop extends Pigeon language in the language layer to support spatiotemporal data types and operations. The indexing layer spatiotemporally divides and loads data across the slave nodes in Hadoop Distributed File System (HDFS). The MapReduce layer provides SpatioTemporalFileSplitter, and SpatioTemporalRecordReader components to exploit the index structures and speed up spatiotemporal operations. The operations layer presently supports three spatiotemporal operations or queries: 1. Join queries 2. Spatiotemporal range queries and 3. Nearest neighbor queries.

The fact behind the best performance of ST-Hadoop is its capability to load the data in HDFS in a way that mimics spatiotemporal index structures so that spatiotemporal queries can have minimal data access to answer the query. ST-Hadoop can be a research vehicle to practitioners, developers and researchers in spatiotemporal big data analytics.

IVI. SPARK BASED SPATIOTEMPORAL BIG DATA SYSTEMS

4.1 STARK

STARK [13] is a framework that designed new classes for integrating temporal and spatial data types, spatiotemporal data partitioners, indices, join, filter, k-Nearest Neighbor (kNN) queries and cluster operators to Apache Spark. A special class called STObject provides two fields: geo to store spatial aspects and time for holding temporal aspects of an object. STARK provides Application Programming Interface (API) involving these classes and Resilient Distributed Dataset (RDD) transformations for spatiotemporal operations. An RDD is partitioned using partitioner and can be indexed. STAK also provides indexing in two modes: live indexing is built for each partition while query execution and persistent indexing is to build and save indexed RDD into HDFS or disk for later use. STARK also provides Piglet by extending Pig Latin for declarative queries on spatiotemporal data.

4.2 DiStRDF

The Resource Description Framework (RDF) data is modeled as statements containing triples (subject, property, object) for interchanging the data over web. It can be queried using a declarative language called SPARQL. Distributed Spatiotemporal RDF (DiStRDF) [14, 15] system is designed based on Spark to do efficient and scalable management for querying spatiotemporal RDF data. DiStRDF has two important modules: DiStRDF Storage Layer and DiStRDF Processing Layer. The storage layer stores encoded dictionary of mapping data in Redis key-value store and encoded RDF data triplets in HDFS. The Processing Layer is based on Spark query engine. It does planning, parsing and execution of SPARQL queries. It distributes 1D encoded RDF triples using spatiotemporal range partitioning and supports indexing of RDF triples using Hilbert and Z-order hashing. However, it supports spatiotemporal point data only.

4.3 Beast

Beast [16] is based on Apache Spark. It offers following five extensible components to provide various features for spatiotemporal big data processing.

i. A set of parallel writers and loaders for spatiotemporal big data that is in various formats such as CSV, Shapefile, GeoTIFF, and GeoJSON. This component also offers scalable spatiotemporal data generators for benchmarking and stress testing.

ii. Spatiotemporal data load balancer and partitioner component offers a set of spatiotemporal data partitioning techniques for grouping spatially related records, creating partitions and distributing them across the executor nodes while balancing the load.

iii. The interactive query processing component facilitate approximate query processing, e.g., Bloom filter, ample, Euler histogram, and point histogram. It builds approximate algorithms such as selectivity estimation and clustering.

iv. A scalable join component provides distributed join algorithms with different optimizations for handling spatiotemporal big data efficiently. It uses rule based optimizer for choosing most appropriate technique to join multiple spatiotemporal datasets together.

v. An exploratory map interface component helps in visualization and analysis of query results or input data using interactive map interface. Beast provides five interface to use its components: RDD API for Scala and Java, SQL API, Interactive Shell, Command-Line Interface (CLI), and Web Interface.

Experiments on Beast using large scale data has proved its scalability and usefulness. It can be used as a research vehicle for conducting research on spatiotemporal big data in various domains.

VI. NoSQL BASED SPATIOTEMPORAL BIG DATA SYSTEMS

5.1 MD-HBase

MD-HBase [17] is an extension of HBase. It has built a multi-dimensional index over HBase. It transforms the multidimensional spatiotemporal data having identifier, latitude, longitude and timestamp into one dimensional space using linearization techniques such as Z-ordering for efficient indexing and processing of multidimensional queries such as range and kNN. MD-HBase creates Quad tree and KD trees on top of the HBase data store. It provides high insert throughput of spatiotemporal big data and ensures high availability and fault tolerance.

5.2 GeoMesa

GeoMesa [18] contains a suite of tools such as HBase, Accumulo, Cassandra, Redis, Kafka and Spark for storing, indexing, querying, transformation, and analysis of spatiotemporal big data. It linearizes the multidimensional data having location and timestamp into one dimensional space using a technique called space-filling curves. It provides a spatiotemporal index on top of the Accumulo, HBase, Google Bigtable and Cassandra databases using GeoHash and timestamps for massive storage of point, line, and polygon data. It supports spatiotemporal indexing techniques such as Z3 and XZ3. It also process real time stream of spatiotemporal data by providing a spatial semantics on top of Kafka.

5.3 JUST

JD Urban Spatio-Temporal (JUST) [19] integrates the power of Spark, GeoMesa, and HBase into one system. Its storage layer uses HDFS and HBase. It introduces two more indexing techniques, XZ2T and Z2T besides Z3 and XZ3 of GeoMesa. Its querying layer called JustQL is similar to SQL and was developed from the scratch. It uses Spark as a query processing engine. JUST reduces memory requirement in the nodes of a cluster by loading only necessary spatiotemporal data into memory. It has improved performance of query through efficient utilization of main memory. Experiments conducted on JUST using one synthetic dataset and two real datasets show that it is more scalable and efficient in query processing.

VII. OTHER SPATIOTEMPORAL BIG DATA SYSTEMS

6.1 AsterixDB

AsterixDB [20] is a powerful full-fledged distributed spatiotemporal big data management system. It uses LSM-based spatiotemporal data storage, and R-tree and B+-tree as indexing techniques. Its query language called AQL uses Hyracks as a query execution engine. It consists of a set of built-in data types including spatial and temporal data. It allows to perform spatiotemporal queries. Currently, SQL++ query language is also supported by AsterixDB for semi-structured data. It has Cluster Controller, Metadata Node Controller, and Nodes Controllers as components. The Cluster Controller is the logical entry point for queries and requests. The Metadata Node Controller offers an access to metadata of AsterixDB. The Nodes Controllers of underlying shared nothing architecture provide an aggregate processing power in AsterixDB.

6.2 DISTIL

DISTIL [21] is a spatiotemporal in-memory system, developed from scratch based on Asynchronous Partitioned Global Address Space (APGAS) programming model. It uses master slave architecture. The master node does coordination of various activities running on the slaves. The Coordinator receives record and redirects them to data updater component running on the slaves. Each slave node stores the location records corresponding to the spatial partitions it hosts. Efficient data partitioning and distributed multi-level spatiotemporal indexing of DISTIL help in enhancing performance of kNN and range queries. A distributed query execution plan is generated and executed in each slave. The partial results generated at each slave are aggregated to generate the final result.

DISTIL aggregates all the main memory of the slaves in the cluster using the APGAS model to have low latency data processing. DISTI has two persistent stores: A Fast Local store at each slave and a Global store based on HDFS. . High rate of insertion and updates of record is achieved in DISTIL by using LSM Tree based key-value store called LevelDB as a local data store. DISTIL executes an offline process periodically to transfer data from the Local store to the Global store.

Generally, data processing systems built from scratch for a specific purpose, such as DISTIL for spatiotemporal data processing, can achieve good performance. But it is always challenging to develop a full-fledged system from scratch and also it is hard to use them as a general-purpose systems.

6.3 RAPIDS AI and TransBigData

CPU is a bottleneck in CPU based in-memory systems like Spark for big data processing. It is due to few cores in CPUs. GPUs contains hundreds of cores, high-speed hardware interconnections and high bandwidth memory. RAPIDS AI [22] is a set of open source software libraries and APIs. It is developed by NVIDIA to leverage GPUs in PySpark by running end-to-end data processing pipelines. The data processing pipelines span from Spark executors to Python workers and finally lands on GPUs without losing execution speed at the cost of serializations, data movement and data conversions. RAPIDS AI also support multi-nodes, multi-GPU deployments to accelerate processing and training deep learning [8,9] algorithms on large datasets.

Apache Arrow framework minimize data serializations and data conversions when data processing pipeline contains various computing frameworks. The RAPIDS AI cuDF API is a DataFrame manipulation library based on Apache Arrow. It accelerates data loading, data filtering, and data manipulation operations in various big data applications.

TransBigData [23] is a Python package developed for spatiotemporal big data processing in transportation domain. It provides flexible, clean, easy to use, and efficient API for Data Quality, Data Preprocess, Data Gridding, Data Aggregating, Trajectory Processing, Data Visualization and Basemap Loading.

VII COMPARISION OF SPATIOTEMPORAL BIG DATA SYSTEMS

All the Spatiotemporal Big data Systems reviewed in this paper are compared with respect to underlying platform, data types supported, partitioning and indexing techniques used, query language and queries supported. The results of comparison are provided in Table 2.

Table 2: Comparison of Features of Spatiotemporal Big Data Systems

S.No	System Name	Underlying Platform	Data Types	Partitioning	Indexing	Query Language	Supported Queries
1	ST-Hadoop [12]	Hadoop	STPoint, Time, Interval	Time-Slice, Data-Slice	Two-Level L1: Temporal L2: Spatial	Extended Pigeon	Range, Join
2	STARK [13]	Spark	STObject	Fixed-Grid, Binary-Space	R-Tree	Piglet	kNN, Join
3	DiStRDF [14, 15]	Spark	Point, Timestamp	Range-Partition	Hilbert Hash Z-order Hash	SPARQL	Range, Join
4	BEAST [16]	Spark	Point, Line, Polygon, Tmestamp	R*-Grove	R*-Tree	SparkSQL	Range, Join
5	MD-HBase [17]	HBase	Point, Timestamp	Range-Partition	Quad-Tree, KD-Tree	-	Range, kNN
6	GeoMesa [18]	Accumulo	Point, LineString, Polygon, Timestamp	Spatial, Temporal, Attribute	Z2 and XZ2, Z3 and XZ3,. id and attr	CQL	Range
7	JUST [19]	HBase	Geom, Timestamp, ST_Series, T_Series	-	Z2 and XZ2, Z3 and XZ3, Z2T and XZ2T	JustQL	Range, kNN
8	AsterixDB [20]	Hyracks	Point, Line, Polygon, Circle, Rectangle, Date, Time, Interval, Duration.	Hash	B+-Tree, R-tree	AQL, SQL++	Range, Join
9	DISTIL [21]	APGAS	Point, Timestamp	RRR, MDR	Multi-Level L1: Quad-Tree L2: Spatial L3: Temporal	-	Range, kNN
10	RAPIDS AI and TransBigData [22, 23]	SPARK	Point, LineString, Timestamp	-	-	Python API	Range

VIII. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

Due to the availability of voluminous spatiotemporal data and rate at which it is getting accumulated, and utility value of discovered knowledge in various application domains, plenty of research and development have been done in the field of spatiotemporal data processing during last few years. This review article has discussed recent progress in distributed spatiotemporal big data processing systems which are Hadoop-based, Spark-based, NoSQL-based, or other platforms. The systems are compared with respect to storage, processing and performance features. All these systems have been in research and needs to evolve into full-fledged spatiotemporal big data systems.

There will be a demand in near future to have more support for visualizing spatiotemporal data on web platforms. Also, there is a need for more research and development to add more features for processing spatiotemporal big data and also conduct experiments using GPUs. The future big data systems are likely to be more cloud-native and requires machine learning and deep learning models to process spatiotemporal big data.

The research community needs to focus at least following five areas for enhancing performance and utility of spatiotemporal big data processing systems in near future: (i) visualization (ii) indexing (iii) mining (iv) GPU (v) Emerging technologies such as Artificial Intelligence, Machine Learning, Neural networks and deep learning to realize intelligent spatiotemporal big data processing systems. The information provided in this review article will be useful for developers, practitioners, and researchers to work in the area of spatiotemporal big data analytics.

REFERENCES

[1]. Ali Davoudian, Liu Chen and Mengchi Liu. 2019. A Survey on NoSQL Stores. ACM Computing Surveys. 51(2):1-43. DOI: https://doi.org/10.1145/3158661

[2]. R. Tian, H. Zhai, W. Zhang, F. Wang and Y. Guan. 2022. A Survey of Spatio-Temporal Big Data Indexing Methods in Distributed Environment. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 15:4132-4155. DOI: https://doi.org/10.1109/JSTARS.2022.3175657

[3]. Junmei Xu , Bin Chen , and Liying Sun. 2022. Big Data Storage Index Mechanism Based on Spatiotemporal Information Cloud Platform. Hindawi Security and Communication Networks. DOI: https://doi.org/10.1155/2022/6774821

[4]. Alam, Md. Mahbub, Luís Torgo and Albert Bifet. 2022. A Survey on Spatio-temporal Data Analytics Systems. ACM Computing Surveys. DOI: https://doi.org/10.48550/arXiv.2103.09883

[5]. Yang, Lina, Zhaoting Ma, Lining Zhu, and Li Liu. 2019. Research on the visualization of spatio-temporal data. *IOP Conference Series: Earth and Environmental Science.* 234(1):012013. DOI: https://iopscience.iop.org/article/10.1088/1755-1315/234/1/012013/pdf

[6]. Arun Sharma, Zhe Jiang and Shashi Shekhar. 2022. Spatiotemporal Data Mining: A Survey. arXiv - CS - Distributed, Parallel, and Cluster Computing. arXiv:2206.12753. DOI: https://doi.org/10.48550/arXiv.2206.12753

[7]. Nan Gao, Hao Xue, Wei Shao, Sichen Zhao, Kyle Kai Qin, Arian Prabowo, Mohammad Saiedur Rahaman and Flora D. Salim. 2022. Generative Adversarial Networks for Spatio-Temporal Data:A Survey. ACM Transactions on Intelligent Systems and Technology. 13(2):1–25. DOI: https://doi.org/10.1145/3474838

[8]. S. Wang, J. Cao and P. S. Yu. 2022. Deep Learning for Spatio-Temporal Data Mining: A Survey. IEEE Transactions on Knowledge and Data Engineering. 34(8):3681-3700. DOI: https://doi.org/10.1109/TKDE.2020.3025580

[9]. Anand Gupta, Hardeo Kumar Thakur, Ritvik Shrivastava, Pulkit Kumar and Sreyashi Nag. 2017. A Big Data Analysis Framework Using Apache Spark and Deep Learning, *IEEE International Conference on Data Mining Workshops (ICDMW)*. 9-16. DOI: https://doi.org/10.1109/ICDMW.2017.9

[10]. A. Eldawy and M. F. Mokbel. 2015. SpatialHadoop: A MapReduce framework for spatial data. *IEEE 31st International Conference on Data Engineering*. 1352-1363. DOI: https://doi.org/10.1109/ICDE.2015.7113382

[11]. Zhou Huang, Yiran Chen, Lin Wan and Xia Peng. 2017. GeoSpark SQL: An Effective Framework Enabling Spatial Queries on Spark. International Journal of Geo-Information. 6(9): 285. DOI: https://doi.org/10.3390/ijgi6090285

[12]. Louai Alarabi, Mohamed F. Mokbel, and Mashaal Musleh. 2018. ST-Hadoop: A MapReduce Framework for Spatio-Temporal Data. Geoinformatica. 22(4): 785–813. DOI: https://doi.org/10.1007/s10707-018-0325-6

[13]. Hagedorn, S., Götze, P. and Sattler, K.U. 2017. The STARK Framework for Spatio-Temporal Data Analytics on Spark. Datenbanksysteme für Business, Technologie und Web (BTW 2017). 123-142. DOI: https://dl.gi.de/20.500.12116/679

[14]. Panagiotis Nikitopoulos, Akrivi Vlachou, Christos Doulkeridis, and George A. Vouros. 2018. DiStRDF: Distributed Spatio-temporal RDF Queries on Spark. *EDBT/ICDT Workshops*. DOI: http://ceur-ws.org/Vol-2083/paper-19.pdf

[15]. Nikitopoulos, P., Vlachou, A., Doulkeridis, C. and George A. Vouros. 2021. Parallel and scalable processing of spatio-temporal RDF queries using Spark. Geoinformatica. 25:623–653. DOI: https://doi.org/10.1007/s10707-019-00371-0

[16]. Ahmed Eldawy, Vagelis Hristidis, Saheli Ghosh, Majid Saeedan, Akil Sevim, A.B. Siddique, Samriddhi Singla, Ganesh Sivaram, Tin Vu and Yaming Zhang. 2021. Beast: Scalable Exploratory Analytics on Spatio-temporal Data. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3796–3807. DOI: https://doi.org/10.1145/3459637.3481897

[17]. S. Nishimura, S. Das, D. Agrawal, and A. E. Abbadi. 2011. MD-HBase: A Scalable Multi-dimensional Data Infrastructure for Location Aware Services. *IEEE 12th International Conference on Mobile Data Management*. 1:7–16. DOI: https://doi.org/10.1109/MDM.2011.41

[18]. James N. Hughes, Andrew Annex, Christopher N. Eichelberger, Anthony Fox, Andrew Hulbert, and Michael Ronquest. 2015. GeoMesa: a distributed architecture for spatio-temporal fusion. Procedings of SPIE 9473, Geospatial Informatics, Fusion, and Motion Video Analytics. DOI: https://doi.org/10.1117/12.2177233

[19]. Li, R., He, H., Wang, R., Huang, Y., Liu, J., Ruan, S., He, T., Bao, J., and Zheng, Y. 2020. JUST: JD Urban Spatio-Temporal Data Engine. *IEEE 36th International Conference on Data Engineering (ICDE)*, 1558-1569. DOI: https://doi.org/10.1109/ICDE48307.2020.00138

[20]. Sattam Alsubaiee, Yasser Altowim, Hotham Altwaijry, Alexander Behm, Vinayak Borkar, Yingyi Bu, Michael Carey, Inci Cetindil, Madhusudan Cheelangi, Khurram Faraaz, Eugenia Gabrielova, Raman Grover, Zachary Heilbron, Young-Seok Kim, Chen Li, Guangqiang Li, Ji Mahn Ok, Nicola Onose, Pouria Pirzadeh, Vassilis Tsotras, Rares Vernica, Jian Wen, and Till Westmann. 2014. AsterixDB: A Scalable, Open Source BDMS. Proceedings of VLDB Endow. 7(14): 1905–1916. DOI: https://doi.org/10.14778/2733085.2733096

[21]. Maria Patrou, Md Mahbub Alam, Puya Memarzia, Suprio Ray, Virendra C. Bhavsar, Kenneth B. Kent, and Gerhard W. Dueck. 2018. DISTIL: A Distributed in-Memory Data Processing System for Location-Based Services. *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. Association for Computing Machinery. 496–499. DOI: https://doi.org/10.1145/3274895.3274961

[22]. Aguerzame, A., Pelletier, B. and Waeselynck, F. 2019. GPU Acceleration of PySpark using RAPIDS AI. *Proceedings of the 8th International Conference on Data Science, Technology and Applications – ADITCA*. 437-442. DOI: https://doi.org/10.5220/0008191404370442

[23]. Qing Yu and Jian Yuan. 2022. TransBigData: A Python package for transportation spatio-temporal big data processing, analysis and visualization. The Journal of Open Source Software..7(71): 4021. DOI: https://doi.org/10.21105/joss.04021