EAI Endorsed Transactions Preprint Research Article/Editorial

A method for integrating GIS and big data platforms

Hong Anh Le[1,*]

[1]Faculty of Information Technology, Hanoi University of Mining and Geology, Duc Thang, Bac Tu Liem, Ha Noi, Viet Nam

Abstract

Geographic Information System (GIS) has been played an important role in many applications of our daily life since 1970. Recently, with the rapid development of new technologies, earth's data increases explosively. Many studies have been proposed to extend big data platforms with spatial data storage and processing. GIS users, however, still need a method to work with a large of data sets with traditional tools. This paper proposes a method to integrate ArcMap with Apache Hadoop and its ecosystem. The method has two phases including database creating and querying in Apache Hive. There are two tools following the proposed method are developed for illustration purpose. The experiment results on a data set of taxi trips in a year show that the method impressively improves the query performance.

Received on 01 July 2021; accepted on 09 July 2021; published on 14 July 2021

Keywords: ArcMap, ArcGIS, big data, Apache Hive, Apache Hadoop

doi:10.4108/eai.14-7-2021.170293

1. Introduction

Geographic Information System (GIS) is a system for storing and analysing Earth's data that became widely familiar in the '70s. Since then, it has been played an important role and used in many applications of everyday life. Various types of data sources can be ingested into a GIS producing maps with many data layers. Users are possible to make deep analysis within an interested area and over the period of time using GIS. For this reasons, a lot of tools which are both open source and commercial have been created to support users to dive into GIS. Among them, AcrGIS is one of the most popular commercial GIS solutions provided by ERSI [3]. AcrGIS includes many applications which allows users to work with GIS. GIS data also grows exponentially in size that requires GIS frameworks and tools to adapt. Therefore, researchers have been working out to leverage GIS with big data technologies.

Big data is a recent emerging term that describes the very large volume and complex data to process with traditional systems. The definition of Big Data comes with 5Vs (volume, velocity, veracity, variety, and value). Because the source are varied from structure to unstructured data with huge amount of volume and generated at fast speed, therefore, appropriated storage and processing platforms such as Apache Hadoop, Apache Spark [2] and their ecosystem components are the need. Apache Hadoop and Spark are open source platforms originated by Apache Foundation Organization consisting of components for big data storage, processing, and analysis. The advancements of such technologies motivate the awareness of big data in the next generation of GIS. Many researchers have been dedicated for combining and integrating big data technologies and GIS. Peng Yue *et al.* [11] introduced the term "BigGIS" and several key considerations of the development. Research work [6] has been investigated the extension for spatial processing in Hadoop. Jia Yu *et al.* [10] introduced which is an in-memory computing framework for large-scale spatial data processing. Dong Xi *et al.* [9] presented Simba (Spatial In-Memory Big data Analytic) offering large-scale in-memory spatial queries. Even though, several remarkable results are achieved. There is a need for traditional GIS desktop tools such as ArcMap to interact with big data platforms. It will benefit GIS desktop users with the pros of new technologies while still make use of the classic tools. In the same direction, Ersi's team initially developed a set of tool for processing spatial data in Hadoop. It, however, still requires a lot of complex skill with Hadoop that is unfamiliar with

*Corresponding author. Email: lehonganh@humg.edu.vn

GIS users. To complement this, the article presents an approach to seamlessly integrate ArcMap with Apache Hive. The contributions of the paper are (i) a method to integrate ArcMap queries with Apache Hive queries seamlessly; (ii) a set of ArcMap python tools that create and queries big data in Apaache Hive.

The article is structured as follows. Section 3 comes with background of ArcMap and Apache Hive. Followed by section 2 summarizing the related work. The proposed integration approach is presented in section 4. The development of the tool sets of Taxi trips case study are described in Section 5. Finally, section 6 concludes the article and presents the future work.

2. Related Work

Ersi already provided a set of tools of GIS extension for Hadoop [4]. Our tool is also developed based on their work. The difference is that this paper proposed a generic method for integration with detailed steps and more automatic.

Shaohua *et al.* [8] proposed and developed an integrated GIS plaform with many other latest big data technologies called SuperMap GIS. The platform is integrated and compatible with many powerful platforms such as Spark, Kafka, etc.. But this product is not an open source one and belongs to SuperMap company. They only provide the SuperMap GIS for 90 trial days.

Lopez Vega, M.A *et al.* [7] developed a near real-time environment monitoring system that based on GEOS-R satellite imagery. The system focused on storage satellite imagery data rather than processing.

Ahmed Eldawy and Mohamed F. Mokbel [6] proposed a spatial extension of Hadoop named Spatial-Hadoop. It is an open source platform providing native support for spatial data types and operation. It adapts several spatial structures such as Grid, R-tree, and R+tree. In the same direction, Ablimit Aji *et al.* [5] proposed Hadoop-GIS, a large scale warehouse for spatial data, that supports multiple types of spatial queries on MapReduce.

3. Backgrounds

3.1. GIS and ArcGIS

GIS has three major components including data, hardware, and software. A GIS application allows users to work with digital maps, create new data layer to customize the maps, and analyze the spatial information. The common functionality of GIS is associating non-geographical information with places locations. There are two types of GIS data such as vector and raster data. The former consists of three types: point, line, and polygon data. The later has three types of data sets such as imagery, spectral, and thematic data. Due to huge requests of using GIS, there a bunch of both commercial and open source tools and platforms. ArcGIS is a world-wide popular solution GIS of Ersi. They provides a lot of products and utilities for GIS users with various technologies such as desktop, web, or cloud computing. Ersi also has released several new products also adapts to big data technologies. The traditional tool such as ArcMap, however, is still familiar with many users. As mentioned in the previous section, ArcMap users feel difficult to work with big data sets in ArcGIS desktop environment. In order to make ArcGIS desktop extension, Ersi allows developers to implement ArcMap toolbox with ModelBuilder and Python. The template for creating Python toolbox is described as the following snippets.

```
import arcpy
class Toolbox(object):
    def __init__(self):
class Tool(object):
    def __init__(self):
        self.label = "Tool"
        self.description = ""
        self.canRunInBackground = False
    def getParameterInfo(self):
        """Define parameter definitions"""
        params = None
        return params
    def isLicensed(self):
        return True
    def updateParameters(self, parameters):
        return
    def updateMessages(self, parameters):
    def execute(self, parameters, messages):
        """The source code of the tool."""
        return}
```

3.2. Apache Hadoop and Apache Hive

Apache Hadoop is an open source framework provides availability for distributed processing large data sets across single to thousands of nodes[1]. Hadoop was created by Nutch and a part of Apache Lunce. After implementing NDFS and MapReduce, in 2008, it became an independent one and rapidly reached the Apapche projects top-level in 2010. The basic architecture of Apache Hadoop V2.0 is illustrated in Figure 1. It has three main components including HDFS, MapReduce, and YARN.

- HDFS is designed for distributed file storage
- MapReduce provides features for parallel processing
- YARN stands for Yet Another Resources Negotiation that consists of ResourceManager and NodeManager. The former allocates resources between

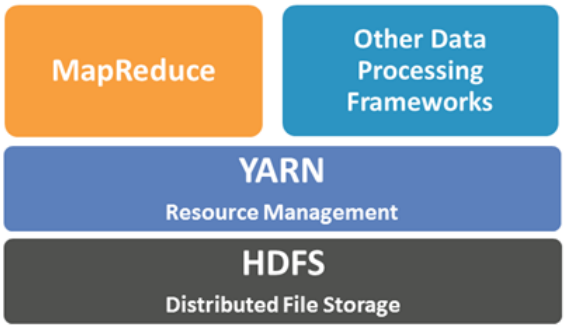

Figure 1. Hadoop V2.0 architecture

all application, the latter monitors the resource usage of the containers and report to ResourceManager.

Apache Hive is an open source warehouse to process structured data in Hadoop. It was initially developed by Facebook, then transferred to ASF. Hive stores schema in a database, while processes data in HDFS. It supports query language which is similar to SQL called HiveQL. The basic architecture of Apache Hive is illustrated in Figure 2.

4. Integration between ArcMap and Hive

This section presents the approach of integration between ArcMap and Apache that allows ArcMap users execute a query over a big data set using the Hive engine running in a Hadoop cluster. The overview of the approach is illustrated in Figure 3.

The system will consist of two components including ArcMap desktop that handle spatial analysis tasks and a Hadoop cluster with Hive installed for storing and processing data. The proposed approach allows to separate GIS features and big data processing. It provides a transparency to ArcMap users and does not require them to manually do the complex tasks such as writing Hive scripts. These scripts are generated automatically. After execution in Hadoop, the query results are sent back ArcMap in form of JSON data file. The integration reuses advancement of distributed storage and processing in Hive because this feature in ArcMap software is limited. GIS users then just only have to handle the very smaller data set results returned from Hadoop cluster.

In order to implement the model, the paper makes use of the method extending ArcMap plugins as toolboxes. Figure 4 shows the detailed steps that integrate a AcrMap toolbox to Hive engine to query big data sets. The integration consists of two phases such as establishing Hive database and making Hive queries.

In order to create Hive database from the big data set, we need to follow the steps below

1. Connecting to Hadoop cluster using configuration file.
2. Generating HiveQL script for creating database tables from a data source.
3. Executing generated HiveQL scripts in Hive CLI.
4. Hive Engine execute script with MapReduce tasks.

This phase aims to automatically put data to Hadoop cluster and execute HiveQL creating database scripts. The second phase's objective is generating HiveQL query scripts from ArcMap toolbox, the result of the query execution is pulled to ArcMap in form of JSON file. This phase consists of the following steps.

1. Connecting to Hadoop cluster using configuration file.
2. Generating HiveQL script for creating database tables from a data source.
3. Executing generated HiveQL scripts in Hive CLI
4. Hive Engine execute script with MapReduce tasks.

5. A case study: Tools for taxi trips queries

5.1. Data Sources

In this example, we need to use ArcMap to analyze data and visualize the results over a big data set of taxi trips in New York in 2013. The monthly data is stored in a CSV file that has approximately 2GB in size and around few millions records. The detailed information of these data files are given in Table 1. The total size of the data source is approximately 23GB with more than 20 millions records. The data with such huge volume will cause ArcMap to work very slow when loading and filtering the data. To solve this, we follow the proposed method to develop a integration tool with Hadoop and Hive.

The data file contains some specific fields as follows

- pickup_datetime: denotes pick up time of the trip.
- dropoff_datetime: drop off time of the trip.
- trip_time_in_secs: duration of the trip counted in seconds
- trip_distance: distance of the trip
- pickup_longitude: longitude of pick up location of the trip.
- pickup_latitude: latitude of pick up location of the trip.

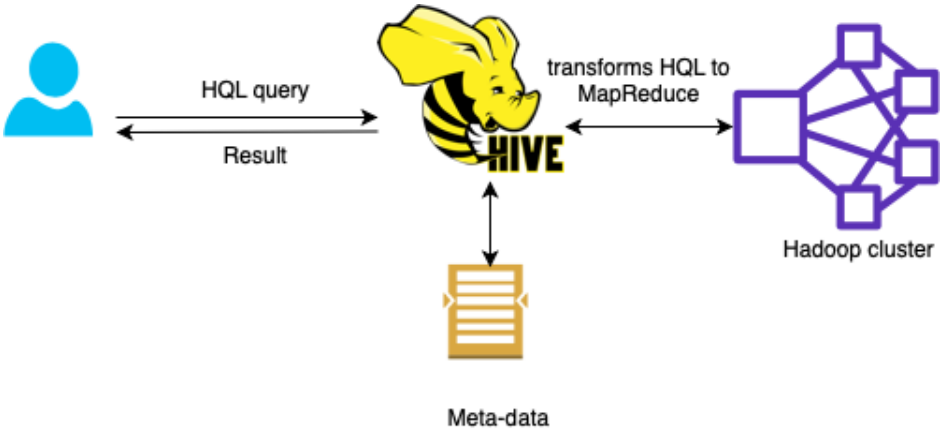

Figure 2. Apache Hive architecture

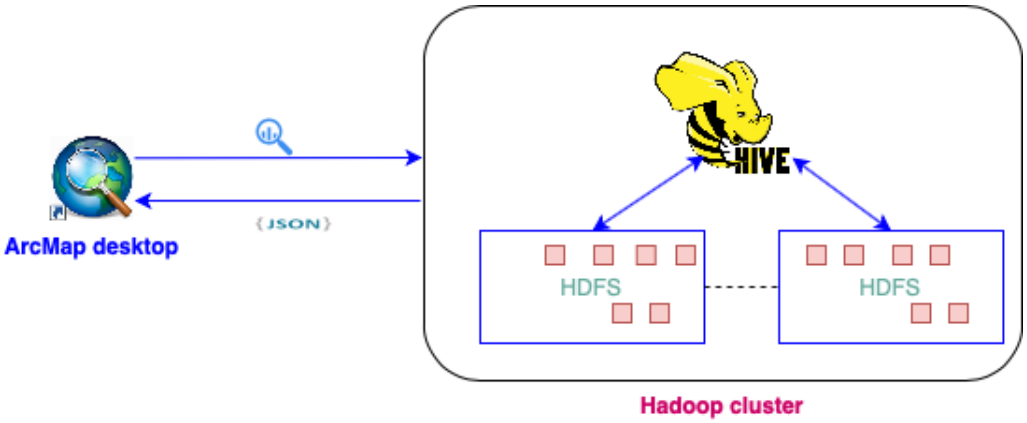

Figure 3. Integration between ArcMap and Hive

Table 1. Taxi trips data in 12 months

File name	Size	No of Records
trip_data_1.csv	2.29 GB	14776615
trip_data_2.csv	2.16 GB	13990176
trip_data_3.csv	2.44 GB	15749228
trip_data_4.csv	2.34 GB	15100468
trip_data_5.csv	2.37 GB	15285049
trip_data_6.csv	2.23 GB	14385456
trip_data_7.csv	2.14 GB	13823840
trip_data_8.csv	1.95 GB	12597109
trip_data_9.csv	2.19 GB	14107693
trip_data_10.csv	2.33 GB	15004556
trip_data_11.csv	2.23 GB	14388451
trip_data_12.csv	2.16 GB	13971118

- dropoff_longitude: longitude of drop off location of the trip.
- dropoff_latitude: latitude of drop off location of the trip.

In order to implement the proposed method, we setup a laptop computer with CPU I7, 4 Cores, 16GB RAM installed ArcGIS desktop and a Linux computer with 8GB RAM, 4 Cores to run Hadoop and Hive.

5.2. Implementation and Results

Toolbox for creating and import database in Hive. This tool contains two input parameters indicating the table name in Hive and path to the data source file. The output the of tool is a Hive script creating table for storing data. The flow is described in detail as follows

- Connect to HDFS server using configuration file.
- If we can connect from ArcMap, then create file for HSQL script.
- Construct HiveQL for creating table and loading the data from data source file.
- Start running PyHive connecting to Hive engine in the toolbox.
- Execute HiveQL scripts using PyHive

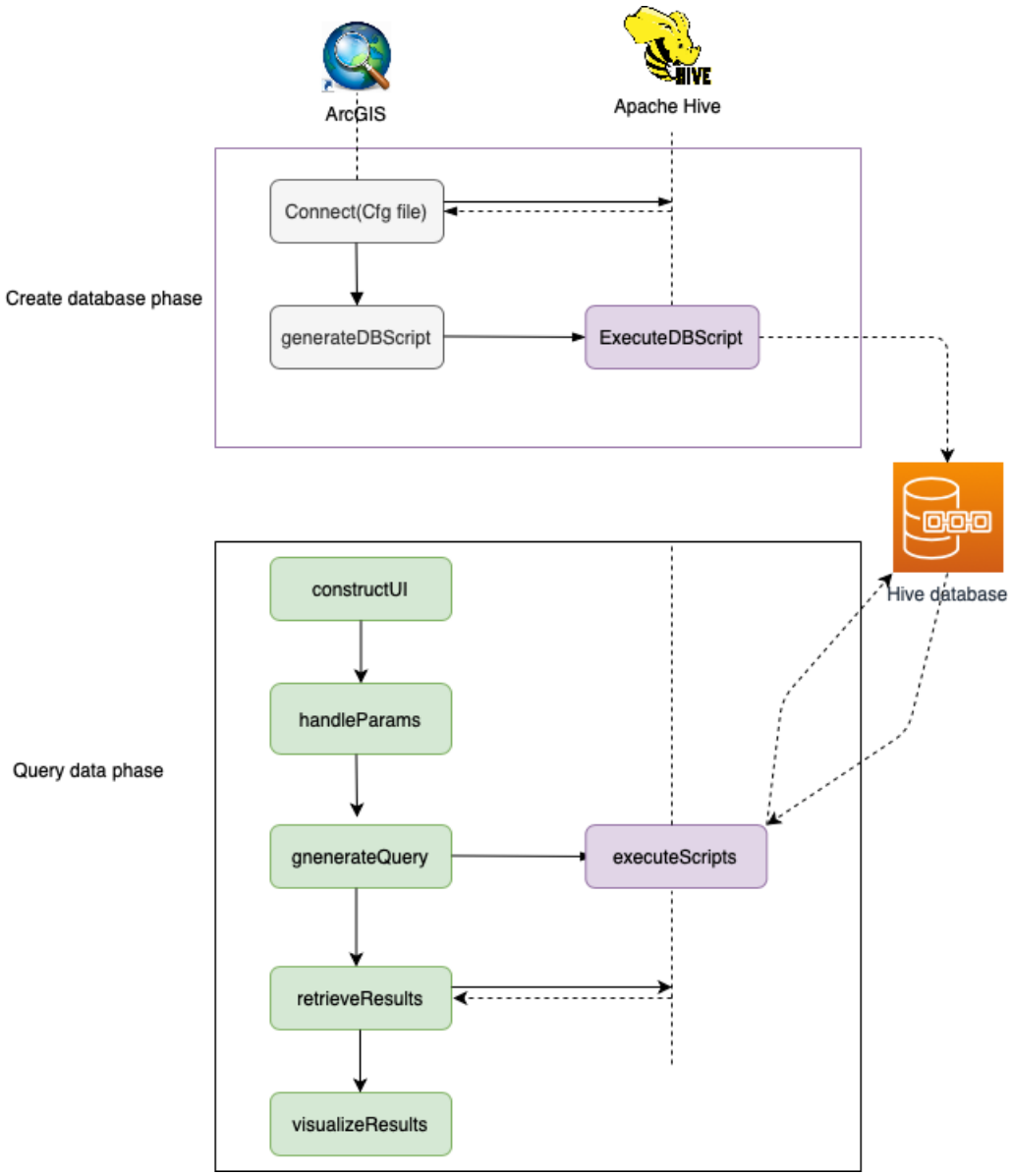

Figure 4. Steps integrating ArcMap toolbox with Hive engine

Toolbox for querying data. This toolbox is designed for specific queries as follows

- Top *N* furthest distance trips
- Top *N* shortest distance trips
- Top *N* largest duration trips
- Top *N* smallest duration trips

The process flow is described in detail as follows

- Connect to HDFS server using configuration file.
- If we can connect from ArcMap, then get input parameters from GUI of toolbox and do the next steps.
- Create HiveQL for querying the data with corresponding input parameters
- Start running PyHive connecting to Hive engine in the toolbox.
- If the query returns results successfully, then export the query result to a JSON file stored in HDFS.
- ArcMap toolbox copies JSON file from HDFS cluster to visualize the result.

Figure 7 shows the developed toolbox for querying data that runs in the ArcMap.

Results. In order to evaluate the developed tool, we have evaluated with 02 data set. One data set is

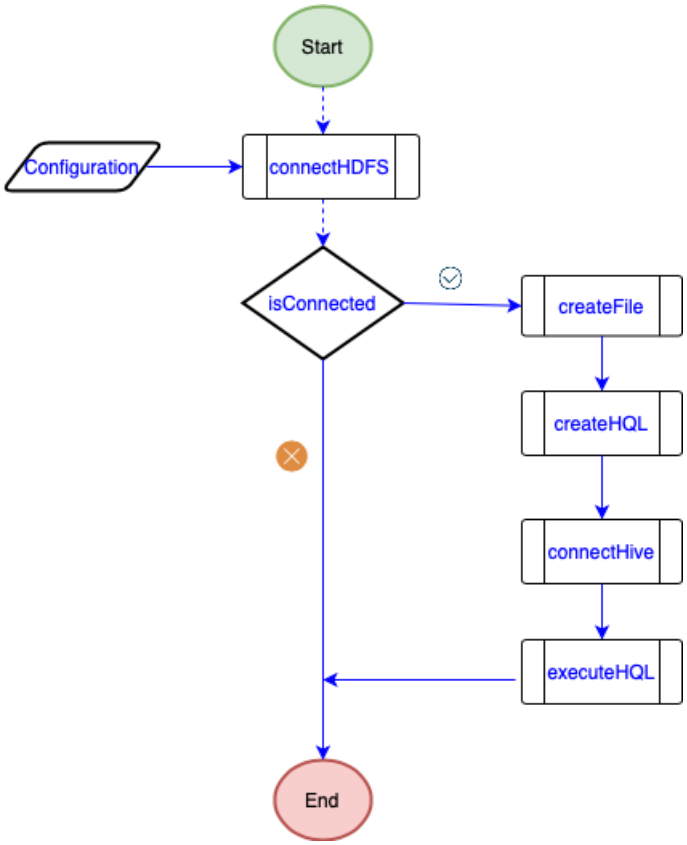

Figure 5. Steps integrating ArcMap toolbox with Hive engine

data of trips in December and one is the whole year log. Figure 8 shows the results when visualizing 100 furthest distance trips in December and the whole year.

ArcMap alone takes approximately around 2 hours to complete processing data amount in December while the developed toolbox completed with under 10 minutes.
With approximately 26GB data of the whole year, ArcMap finds noway to do the filter because the data is so huge, while the constructed tool complete the task within around 50 minutes if the data source file already exists in HDFS.

6. Conclusions

This paper proposes a method to develop a toolbox for processing big data in a ArcGIS desktop tool. It provides detailed steps to implement ArcMap toolbox connecting to Hive server engine. Two toolboxes have been constructed for the illustration purpose. It shows that the tool improves the performance of queries around ten times with a big data set. The experiment result, however, is still limited because we have not used a Hadoop cluster for computing. Moreover, the query structure is still simple with filtering a column in a table. We intend to work further to make the tool more

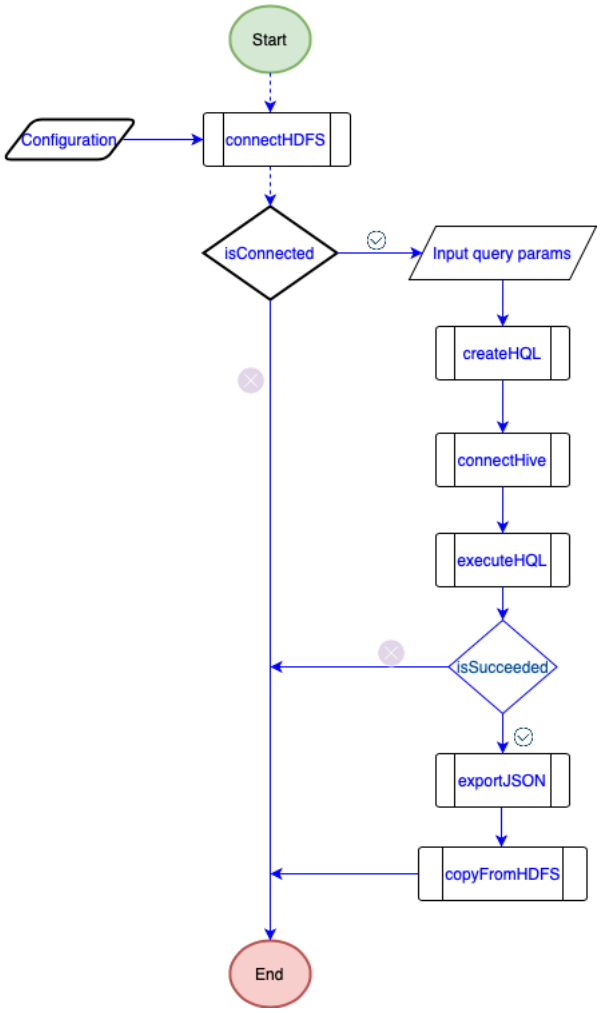

Figure 6. Implementation flow of Query Toolbox

powerful with more complex queries. Hive functions will also be take into account for querying data.

Acknowledgement

This work is supported by the project no. CT.2019.01.05 granted by Ministry of Education and Training (MOET). Thanks to the contribution of Nguyen Thai Dai for his experiment work.

References

[1] Apache Hadoop. https://hadoop.apache.org, 4/2021.
[2] Apache Spark. https://spark.apache.org/, 4/2021.
[3] Arcgis product. https://www.esri.com/en-us/arcgis/about-arcgis/overview, 4/2021.
[4] Hadoop GIS tool. https://esri.github.io/gis-tools-for-hadoop, 4/2021.
[5] A. Aji, F. Wang, H. Vo, R. Lee, Q. Liu, X. Zhang, and J. Saltz. Hadoop GIS. *Proceedings of the VLDB Endowment*, 6(11):1009–1020, Aug. 2013.
[6] A. Eldawy and M. F. Mokbel. SpatialHadoop: A MapReduce Framework for Spatial Data. In *31st IEEE*