USING SYSTEMS OF PARALLEL AND DISTRIBUTED DATA PROCESSING TO BUILD HYDROLOGICAL MODELS BASED ON REMOTE SENSING DATA

A. A. Kolesnikov [1] *, P.M. Kikin [2], E.A. Panidi [2], A.G. Rusina [3]

[1] Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation - alexeykw@yandex.ru
[2] Dept. of Geomatic Engineering, Saint Petersburg State University, St.Petersburg, Russian Federation - it-technologies@yandex.ru, panidi@ya.ru
[3] Novosibirsk State Technical University, Novosibirsk, Russian Federation - rusina@corp.nstu.ru

KEY WORDS: Distributed DBMS, Distributed Processing, Remote Sensing, Raster Data, Climatic Data.

ABSTRACT:

The article describes the possibilities and advantages of using distributed systems in the processing and analysis of remote sensing data. The preparation and processing of various types of remote sensing data (multispectral satellite images, values of climatic indicators, elevation data), which will then be used to build a simulation model of a hydroelectric power plant, was chosen as the basic task for testing the chosen approach. The existing approaches with distributed processing of spatial data of various types (vector cartographic objects, raster data, point clouds, graphs) are analyzed. The description of the developed approach is given and the rationale for the choice of its components is made. The preprocessing operations that were performed on the used raster data are described. An approach to the problems of raster data segmentation based on libraries for distributed machine learning is considered. Comparison of the speed of working with data for various algorithms of machine learning and processing is given.

1. INTRODUCTION

Geospatial and remote sensing data, due to their very large volume, variety and speed of updating, are one of the main elements of the big data concept. Traditional approaches use the power of computing stations to process data, but at the same time they can only scale vertically (which is always costly and the capabilities are severely limited by the hardware platform) and therefore at some point physically cannot cope with the continuous growth of the volume of processed data [7-9]. This problem is most often solved with the help of parallel and distributed processing technologies, which implement the simultaneous processing of each of the parts of the entire data set on a separate node and the combination of intermediate results into the final one [11–13]. There are many tools for parallel and distributed data processing and analysis, but not all of them consider the peculiarities and have the ability to process geospatial data.

The problems facing the authors of the article of predicting spread of tropical diseases, building simulation models of hydroelectric power plants, building databases of natural resource potential require the processing of large volumes of constantly updated remote sensing data on the territory of individual regions and countries in general. Thus, the aim of the study was to build a pipeline for the formation of sets of spatio-temporal indicators for building mathematical models of these objects and processes and predicting their behavior. Today, it is possible to build a distributed storage and processing system using several open-source solutions. So, to do this, it was necessary to analyze the existing open-source software for distributed processing of spatial data, determine their features, advantages, disadvantages, and evaluate the effectiveness of their application on certain datasets.

2. METHODS AND MATERIALS

Today's distributed processing models not only use mechanisms to achieve optimal load balancing, but also provide seamless connectivity between data sources (Panidi and Rykin, 2020). Also, one of the key principles of the most popular distributed processing systems is the maximum encapsulation of internal mechanisms so that users do not need to think about the nuances of parallel processing when developing programs (Barik et al., 2020).

The increasing importance of spatial data leads to the fact that the developers of many distributed computing platforms (Hadoop, Hive, Impala, Spark) extend functionality to handle large-scale spatial data (Venugopal, 2013). Also, at the moment, solutions for streaming spatial data processing are considered as a separate category (Salman et al., 2020). Generally, analyzing the existing directions of using systems and algorithms for distributed processing for spatial data, the following can be distinguished:
- execution of spatial and attribute queries;
- transformation and processing of vector data;
- transformation and processing of raster data;
- conversion and processing of point clouds;
- building indexes for search;
- solving logistic problems;
- generalization for visual display;
- construction of mathematical models.

Execution of spatial and attribute queries implies both the execution of basic queries to search for objects by criteria, as well as a more complex analysis of the spatial relationship of objects. These operations are computationally complex because the search time by criteria increases exponentially with the number of objects, especially if the spatial join operation is

* Corresponding author

present in the selection conditions, which is one of the most computationally complex (Yu et al., 2019).

Operations of modifying vector spatial objects belong to the category of quite simply parallelizable, since processing is carried out for individual objects (even if the original vector object is very large, it is easy to divide it into components). In this case, the only difficulty will be the need to keep track of the topology in relation to neighboring features (Eldawy and Mokbel, 2015). In general, this process usually consists of three sub-processes: partitioning the entire data set, performing calculations on each of the clusters, combining and refining the results. While processing spatial data, it should be considered that the traditional implementation of the second subprocess in distributed systems in the form of dividing data into identical blocks (for example, into a certain number of records in a table or words in a text) is not always suitable, since such data context as geographic location may be lost. and spatial relationships. For this, there are several approaches to preparing data for optimal distributed processing: the use of polygonal regions (Mohan et al. 2011), multilevel graph and tree structures (Andrzejewski and Boinski, 2018; Sierra and Stephens, 2012), regular grids (Sainju et al. , 2018). When choosing the type of division and its parameters, they rely on the density of distribution of objects in space, the maximum number of division blocks, the possibility and amount of spatial overlap (Yoo et al., 2020). From the point of view of implementation examples, one can consider, for example, the definition of spatial association patterns, optimization of spatial queries based on kNN (k-Nearest Neighbor) using MapReduce (Liu t al., 2013; Yoo et al., 2020). More complex is the processing of spatial data presented in the form of heterogeneous models (raster, vector, tabular, etc.), which may also have a temporal component (Werner, 2019). In the case of distributed processing of traditional data, there are usually no problems with the consistency of the final results, but in the case of spatial data, additional operations for correcting the topological structure and semantic coordination of the created and modified objects may be required.

Unlike operations with vector objects, processing and editing of raster data is much easier to parallelize due to their localization during processing. Most operations process one or more pixels located close to each other, which makes these calculations as easy as possible to parallelize with little or no need to solve the problems of mutual matching of data blocks (Boudriki et al., 2019; Sharma et al., 2017; Sharma et al., 2020).

Reducing the cost of light detection and ranging (LiDAR) data, as well as unmanned aerial vehicle (UAV) data, allows to quickly obtain point clouds for large areas of both open areas and indoor areas. But as a result, the problem arises of processing all these data sets in terms of storage and computational complexity. Typical point cloud processing usually involves a number of processes such as point classification, semantic segmentation, ground planes extraction, and feature registration that require complex spatial algorithms (Armstrong, 2014; Venugopal and Kannan, 2013). The organization of parallel processing of this type of spatial data is facilitated by the fact that the most common storage formats are initially focused on a tiled structure, but it requires improvement in terms of creating an additional spatial index for distributed storage (Li et al., 2017).

Existing indexes designed to speed up the processing of spatial queries based on tree structures of various types or regular grids. However, efficient use of these indexes in parallel processing and storage systems is challenging given the different architectures used in each of them, as well as when solving problems with streaming spatial data (Eldawy and Mokbel, 2016). For most cases, the approach is to split a common index into a global index and multiple local indexes distributed across clusters.

Logistic problems are usually solved by applying graph search algorithms. The distributed approach in this case is usually focused on dividing the entire graph into separate clusters based on an adjacency list of edges.

Traditional methods of visualization in geographic information systems are designed to use one computer for processing and further visualization of data, which makes them unsuitable for graphical representation of large volumes of spatial data. Existing algorithms and technologies using graphic processors can significantly speed up rendering processes, but they are still limited by the resources of one computer (there are technologies for parallel processing of three-dimensional images on several video cards installed in one computer, but they are usually limited to 2-3 instances) and cannot scale to a cluster. Existing distributed rendering solutions fall into two categories based on the structure of the image being created: siblings and layered images. In sibling images, the generated raster consists of one data block, which renders the area with a given scale. In multidimensional images, the generated raster consists of a set of image blocks, each of which is a separate scale level (pyramids). Also, each scale, in turn, can be divided into blocks (tiles), allowing users to arbitrarily change the scale and viewpoint (Eldawy and Mokbel, 2016).

It should also be mentioned that in works related to parallel and distributed processing of spatial data, much attention is paid to the implementation in general and the peculiarities of using algorithms on Graphics Processing Units (GPUs) (Andrzejewski and Boinski, 2018; Sainju et al., 2018).

The research methodology consists in comparing the processing speed of various remote sensing data for the same territory using the traditional approach and distributed processing technologies (Çatak and Balaban, 2013; Lv et al., 2010). In the case of distributed processing, options are additionally considered with a different number of nodes and for different stages of data processing. Data processing stages in this case mean the calculation of indices based on spatial channels, terrain characteristics based on elevation data (aspect, slope, curvature), feature engineering for satellite images using dimensionality reduction algorithms (PCA, SVD), down- and up sampling raster images. Additionally, the speed of building distributed machine learning models was estimated, which are focused on the subsequent semantic segmentation of satellite data. The highlighted classes will be hydrographic objects, woodlands and road network.

Based on this goal, the general pipeline was divided into elements responsible for receiving, preprocessing, performing the necessary calculations, generating results based on machine learning models and visualizing them in the form of maps and diagrams.

The territories for which datasets were formed from remote sensing data and vector cartographic base to the territory of Asian Russia (parts of Novosibirsk, Irkutsk, Tomsk, Khabarovsk regions) with a total area of approximately

1,700,000 square kilometers. Also, to compare the results of processing satellite images with the constructed machine learning models, data were taken on the territory of the Republic of the Philippines, Republic of Tajikistan with a total area of about 200,000 square kilometers. These regions were selected because further research on geospatial modeling of natural processes is expected there (all data collection regions are shown in Figure 1).

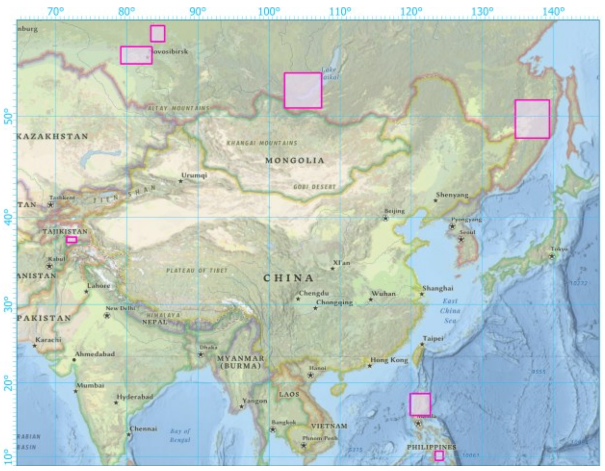

Figure 1. Data collection regions.

The initial data for processing were raster data sets from the Landsat-8, Sentinel-2 and MODIS satellites, the values of meteorological indicators (precipitation intensity, surface temperature) of the World Meteorological Organization, elevation data Alos Palsar, parameters of the Copernicus snow cover (spatial resolution and categories of data sources are presented in Table 1). It is worth noting the variety of source data formats in the form of TIFF, Network Common DataForm (NetCDF), Hierarchical Data Format (HDF5), ESRI Shape, BUFR files. For the study, data were taken from 2014 to 2020.

Category	**Source**	**Resolution (meters)**
Multispectral satellite data	LandSat-8 Sentinel-2	30 20
Snow cover area	MODIS MOD10A1	500
Snow-water equivalent	Copernicus Snow Water Equivalent	5000
Rain intensity	GPM DPR	5000
Elevation data	Alos Palsar Sentinel-1	12.5 20

Table 1. Data sources.

If we talk about the data processing algorithm in general, then in this work it is implemented according to the ETL principle. ETL (extract, transform, load) includes extracting data from the described sources, the necessary transformation operations, and finally loading into the file system or HDFS. HDFS (Hadoop Distributed File System) was used because it is an integral part of Hadoop and the foundation of big data infrastructure. This method is expedient in this case because the initial data are heterogeneous, and it is necessary to apply a lot of transformations to them in order to obtain a single geospace. Data acquisition and processing was implemented based on the combination of Apache Airflow and Apache NiFi functions. The first software is open-source and initially focused on distributed data processing and allows to develop, plan and monitor ETL / ELT processes. The second software is also open-source and is used for routing and transforming data streams, not in terms of scheduling jobs, but to collect data from multiple sources and perform separate steps to process that data. It should also be noted that all parallel processing was performed for small blocks of source images. The base block size was 250 by 250 pixels of the most accurate satellite source (in this case, it was Sentinel-2). This size was taken since pretrained neural networks (in particular U-Net and DeepLabv3) were supposed to be used as one of the semantic segmentation methods, where often the size of the input batch is just a 250 by 250 elements matrix.

Apache Spark software was used to perform calculations on the generated blocks. It is a computing system for parallel data processing on clusters of computers (Hegeman et al., 2018). Unlike the classic Apache Hadoop kernel engine with a two-tier MapReduce concept based on disk storage, Spark uses specialized primitives for recurrent in-memory processing. This makes many computational tasks much faster using Spark. At the moment, Spark is considered the most actively developed open-source tool for solving such problems. Spark supports libraries for tasks ranging from SQL to streaming to machine learning (Spark ML was used in this work).

In addition to directly obtaining all of the above data, it was also required to perform preprocessing operations. Most of these operations related to raster algebra – NDVI and MNDWI indices, terrain characteristics in the form of aspect, slope, curvature and converting raster data blocks to a single resolution. NDVI (Normalized Difference Vegetation Index) - a simple quantitative indicator of the amount of photosynthetically active biomass, calculated from the values in the red and infrared spectrum. MNDWI (Modified Normalized Difference Water Index) using 3 and 6 channels of Landsat 8 and is currently the most common index for identifying surface water objects in satellite imagery.

Thus, the resulting dataset for each pixel of the raster consists of the initial values for spectral channels, additional climatic parameters, previously described indices, values of surface heights and their characteristics. This set of raster values has been expanded to include calculated values using the PCA dimensionality reduction algorithm. PCA (Principal Components Analysis) calculates principal components based on eigenvectors and the values of the covariance matrix of the original data or the singular value decomposition of the data matrix.

After preliminary processing, individual areas of the territories were marked (types of land use, number of cases of infection, water surface area, etc.) to form machine learning models. The generated models were further used to process the entire data set.

When choosing software, we focused on the ability to work based on distributed storage and computing technologies, the map-reduce paradigm, etc.

When choosing machine learning algorithms for semantic segmentation of the obtained datasets, the authors focused on those implementations that support distributed learning. In general, distributed machine learning is divided into two groups: distributed models and distributed data. In the first case, a complex model is distributed among the cluster nodes in order to perform calculations on new data in parallel. For example, in the case of a neural network, layers can be stored separately by nodes in order to quickly train the layer's neurons on a specific node and transmit further, and in the case of a random forest, each tree will be trained and perform predictions on a separate node, in parallel with other trees. In the second case, a separate orchestrator server is allocated in the system architecture, which manages the learning process. Based on the specified parameters, it determines the initial model and delegates to each of the cluster nodes. Each node operates on only one of the blocks of the entire data array, on which it trains the given model. Upon completion of the training process, the nodes send the results to the orchestrator server, which aggregates the weights of all received models. This process can be performed iteratively with an indication of stopping in case of exhaustion of the specified number of training repetitions, or upon reaching the specified accuracy. For the experiments, we took Naïve Bayes, Decision Tree, Random Forest from the Spark ML extension. This library is focused on making it easier to create and use scalable machine learning models and implements a distributed data learning option. Since the authors have experience using PyTorch for processing satellite images, the Deep Speed module was used for scaling. Spark ML also supports distributed model of deep learning, but virtually all are focused on the implementation of Tensorflow.

As a measure of the complexity and comparison of the described implementations, both in data processing and in the construction of machine learning models, the execution time of computations for the same data block was evaluated. For computations using Docker, virtual nodes were formed with configurations in the form of 2G RAM, 20GB external memory, 2GHz computing processor. To assess the influence of the number of elements on the processing speed, clusters were formed, consisting of 1, 3, 6, 8 nodes.

To evaluate the results of segmentation of the used raster data, the standard metric Dice (F1-Measure) was chosen, which implements asymmetric measures of similarity, reflecting the degree of proximity of one object to another.

The results were additionally visualized to assess the quality and interpretation of the models.

3. RESULTS AND DISCUSSIONS

Among the open-source platforms for distributed storage, there is practically no alternative to Apache Hadoop, which has been used. Apache AirFlow was used to build pipelines for loading the initial data. The preprocessing operations (raster algebra) were implemented through RasterFrames and its SuperSet extension, using Apache Spark mechanisms.

After studying existing solutions and performing trial processing experiments, the final pipeline was formed (Figure 2).

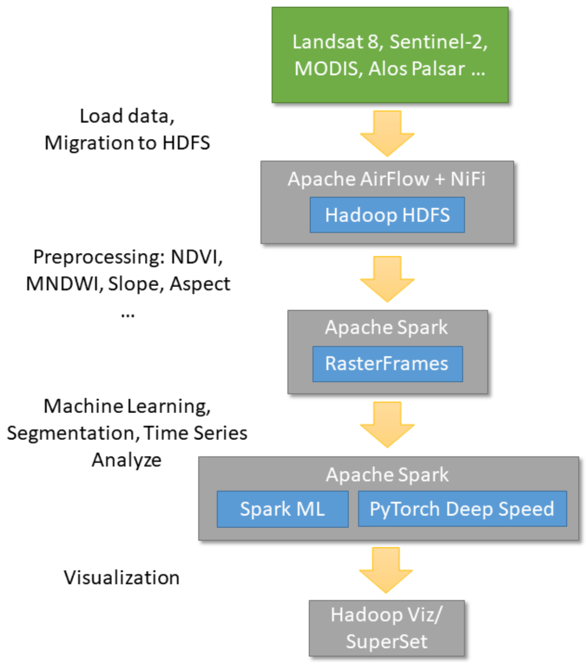
Landsat 8, Sentinel-2, MODIS, Alos Palsar ...
Load data, Migration to HDFS
Apache AirFlow + NiFi
Hadoop HDFS
Preprocessing: NDVI, MNDWI, Slope, Aspect ...
Apache Spark
RasterFrames
Machine Learning, Segmentation, Time Series Analyze
Apache Spark
Spark ML
PyTorch Deep Speed
Visualization
Hadoop Viz/ SuperSet

Figure 2. Processing Pipeline.

Figure 3 shows examples of the results of processing the initial data when calculating indices and the characteristics of the relief. The lower right image shows an example of semantic data segmentation for vegetation classes, hydrography, road network based on a model created using the Random Forest algorithm.

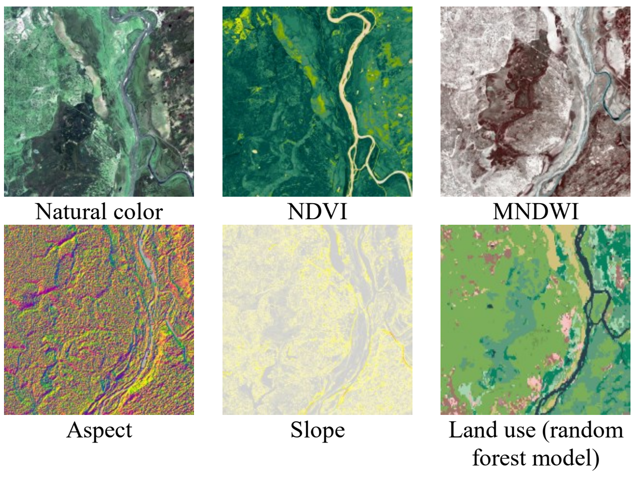

Figure 3. Ground-truth and forecasting water surface boundaries.

Machine learning models were generated in Apache Airflow using an ensemble of the results of the Spark ML and PyTorch algorithms. As part of the study, the HadoopViz and SuperSet software packages were used to visualize the intermediate and final results. From the point of view of hydrological modeling, after segmenting the remote sensing data, a stand-alone model was built that predicts the water surface area based on time intervals for the entire time period under consideration. The result of the model's operation is the area of the water surface and its geometry in a separate section (Figure 4 shows a comparison of the real contours of the river section at different points in time and the constructed forecast).

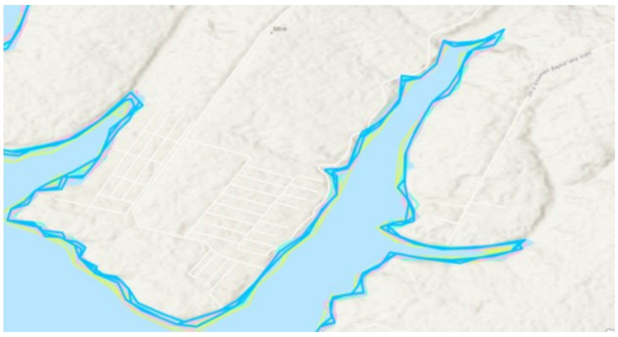

Figure 4. Ground-truth and forecasting water surface boundaries

All the selected tools showed performance gains in the processing of remote sensing data and semantic segmentation of objects on satellite images using machine learning methods. These initial data will later be used in the construction of spatio-temporal models of hydrological objects. Figure 5 shows a relative comparison of the execution time of individual processes in single and parallel versions on the same dataset. The time of processing (or training) at only one node was taken as a unit.

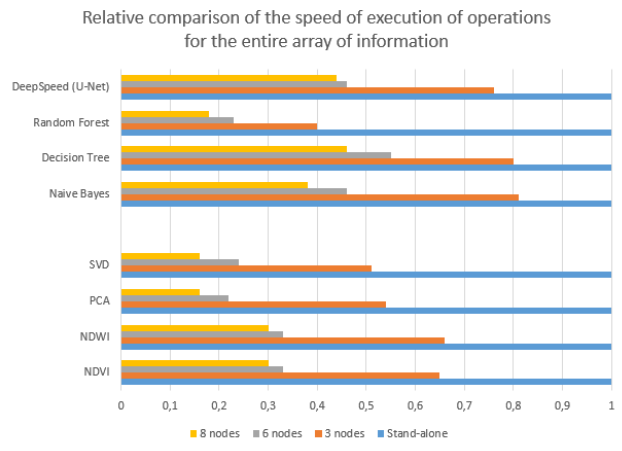

Figure 5. Diagram of relative comparison of the speed of execution of operations for the entire array of information

Analyzing the results, we can say that the existing tools for parallel and distributed data processing allow you to build a full cycle of processing heterogeneous spatial data, including for spatial analysis and machine learning. But it takes a significant amount of time to coordinate all the components and select the processing parameters (affecting the accuracy of the obtained forecast results). It should also be noted that there are practically no (or are outdated) tools for integration with the most popular open-source desktop geographic information systems.

During the experiments, it was noted that the computational performance of the system increases with an increase in the number of processing nodes, but at the same time, an increase in the number of nodes led to an increase in the time spent on data transfer, so it is important to maintain a balance between computing performance and the number of nodes.

It should also be noted that many algorithms for distributed spatial computing do not always have a reliable and multisystem implementation, which makes it difficult to reliably compare different approaches from a practical point of view.

ACKNOWLEDGEMENTS

The article was prepared within the grant for carrying out major scientific projects in priority areas of scientific and technological development under the subprogram "Fundamental scientific research for the long-term development and ensuring the competitiveness of society and the state" of the state program of the Russian Federation "Scientific and technological development of the Russian Federation," project "Social and economic development of Asian Russia on the basis of synergy of transport accessibility, system knowledge of natural resource potential, and expanding area of interregional interactions", agreement with the Ministry of Science and Higher Education of the Russian Federation № 075-15-2020-804 (internal number 13.1902.21.0016).

REFERENCES

Andrzejewski W, Boinski P (2018) Efficient spatial co-location pattern mining on multiple GPUs. *Expert Syst Appl* 93(C):465–483

Armstrong, M. P., 2014. Distributed LiDAR data processing in a high-memory cloud-computing environment. *Annals of GIS*, 20(4), 255-264.

Barik, R. K., Priyadarshini, R., Lenka, R. K., Dubey, H., & Mankodiya, K., 2020: Fog Computing Architecture for Scalable Processing of Geospatial Big Data. *International Journal of Applied Geospatial Research (IJAGR)*, 11(1), 1-20. doi.org/10.4018/IJAGR.2020010101

Boudriki S. B., El Amrani C., Ortiz G., 2019: Adopting the Hadoop Architecture to Process Satellite Pollution Big Data. *International Journal of Technology and Engineering Studies.* 5. doi.org/10.20469/ijtes.5.40001-2.

Çatak, F. Ö. and M. E. Balaban, 2013. A MapReduce based distributed SVM algorithm for binary classification. *Turkish Journal of Electrical Engineering & Computer Science*.

Eldawy A., Mokbel M. F., 2015: Spatial Hadoop: a map reduce framework for spatial data. *IEEE 31st International Conference on Data Engineering,* 1352–1363. doi.org/10.1109/ICDE.2015.7113382

Eldawy, Ahmed & Mokbel, Mohamed. (2016). The Era of Big Spatial Data: A Survey. Foundations and Trends in Databases. 6. 163-273. 10.1561/1900000054.

Hegeman, J. W., Sardeshmukh, V. B., Sugumaran, R., &
Lan H., Zheng X., Torrens P. M., 2018: Spark Sensing: A Cloud Computing Framework to Unfold Processing Efficiencies for Large and Multiscale Remotely Sensed Data, with Examples on Landsat 8 and MODIS Data*, Journal of Sensors,* 2018, 2075057. doi.org/10.1155/2018/2075057

Li, Z. & Hodgson, Michael & Li, Wenwen. (2017). A general-purpose framework for parallel processing of large-scale LiDAR data. *International Journal of Digital Earth*. 10. 1-22. 10.1080/17538947.2016.1269842.

Liu, Y.; Jing, N.; Chen, L.; Wei, X. Algorithm for processing k-nearest join based on r-tree in mapreduce. *J. Softw.* 2013,24, 1836–1851

Lv, Z., Hu, Y., Zhong, H., Wu, J., Li, B., & Zhao, H., 2010. Parallel K-means clustering of remote sensing images based on MapReduce. *Web Information Systems and Mining*, pp. 162-170.

Mohan P, Shekhar S, Shine J, ROgers J, Jiang Z, Wayant N. 2011: A neighborhood graph based approach to regional co-location pattern discovery: a summary of results. *In: Proceedings of the ACM SIGSPATIAL international conference on advances in geographic information systems*, pp 122–132

Panidi E.A., Rykin I.S., 2020: Toward the capabilities of integration of the cloud-based spatial data infrastructures and universal desktop geographic information systems, case study of Google Earth Engine and QGIS. *InterCarto. InterGIS. GI support of sustainable development of territories: Proceedings of the International conference.* Moscow: Moscow University Press, 2020. 26(1). 421–433. doi.org/10.35595/2414-9179-2020-1-26-421-433

Sainju AM, Aghajarian D, Jiang Z, Prasad SK (2018) Parallel grid-based colocation mining algorithms on GPUs for big spatial event data. *IEEE Trans Big Data*. https://doi.org/10.1109/TBDATA.2018.2871062

Salman S., Mariam, Komal & Kitagawa, Hiroyuki & Kim, Kyoungsook. (2020). GeoFlink: A Framework for the Real-time Processing of Spatial Streams.

Sharma T., Shokeen V., Mathur S. 2020: Distributed Approach to Process Satellite Image Edge Detection on Hadoop Using Artificial Bee Colony. *International Journal of Service Science, Management, Engineering, and Technology.* 11. 80-94. doi.org/10.4018/IJSSMET.2020040105.

Sharma, T., Shokeen, V., & Mathur, S., 2017: Distributed Processing of Satellite Images on Hadoop to Generate Normalized Difference Vegetation Index Images. *International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, 1-5.

Sierra R, Stephens CR. 2012: Exploratory analysis of the interrelations between co-located boolean spatial features using network graphs. *Geogr Inf Sci* 26(3):441–468

Vatsavai RR, Ganguly A, Chandola V, Stefanidis A, Klasky S, Shekhar S. 2012: Spatiotemporal data mining in the era of big spatial data: algorithms and applications. *In: Proceedings of ACM SIGSPATIAL international workshop on analytics for big geospatial data*, pp 1–10

Venugopal, V., Kannan, S. (2013). Accelerating real-time LiDAR data processing using GPUs. 2013 *IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 1168-1171, IEEE.

Werner M (2019) Parallel Processing Strategies for Big Geospatial Data. *Front. Big Data* 2:44. doi: 10.3389/fdata.2019.00044

Yoo, J.S., Boulware, D. & Kimmey, D. 2020: Parallel co-location mining with MapReduce and NoSQL systems. *Knowl Inf Syst* 62, 1433–1463. https://doi.org/10.1007/s10115-019-01381-y

Yu J., Zhang Z., Sarwat M.. 2019: Spatial data management in Apache Spark: the GeoSpark perspective and beyond. *Geoinformatica,* 23, 1, 37–78. doi.org/10.1007/s10707-018-0330-9