

UDC 004.75

Data Compression-Aware Performance Analysis of Dask and Spark for Earth Observation Data Processing

Arthur G. Lalayan

Institute for Informatics and Automation Problems of NAS RA, Yerevan, Armenia
National Polytechnic University of Armenia, Yerevan, Armenia
e-mail: arthurlalayan97@gmail.com

Abstract

High-performance computing is a good choice for handling Big Earth Observation data, allowing the processing of the data in a distributed and performance-efficient way using in-memory computing frameworks. The data compression technique reduces the amount of storage and network transfer time and improves processing performance. The article aims to investigate the effectiveness of widely used distributed data processing frameworks in conjunction with lossless data compression techniques, to find the optimal compression method and processing framework for specific earth observation workflows. Normalized Difference Vegetation Index has been evaluated for the territory of Armenia, obtaining data from the Sentinel satellite and considering the supported compression methods to compare the performance of in-memory Dask and Spark frameworks. Experiments show that the Zstandard compression method and the Dask framework are the best choices for such workflows.

Keywords: Earth observation, HPC, Spark, Dask, Distributed computing, Data compression.

Article info: Received 29 January 2022; sent for review 7 February 2023; received in revised form 15 March 2023; accepted 17 April 2023.

Acknowledgement: The research was supported by the Science Committee of the Republic of Armenia and the University of Geneva Leading House by the projects entitled Self-organized Swarm of UAVs Smart Cloud Platform Equipped with Multi-agent Algorithms and Systems (Nr. 21AG-1B052), Remote sensing data processing methods using neural networks and deep learning to predict changes in weather phenomena (Nr. 21SC- BRFFR-1B009), and ADC4SD: Armenian Data Cube for Sustainable Development.

1. Background and Motivation

Earth Observation (EO) satellite data are necessary for environmental monitoring and gathering vital information about various Earth layers [1]. Specifically, EO data are widely used

to monitor the atmosphere including air pollution [2] and temperature [3], the oceans considering sea pollution and ocean acidity [4], and ground, such as deforestation [5] and forest fire [6], as well as to detect climatic changes [7].

To facilitate work with EO data, Australian researchers [8] have provided an open-source Open Data Cube (ODC) [9], which is deployed and widely used by several communities from different countries, including Armenia [10]. Nevertheless, the ODC communities still encounter the Big EO data processing challenge requiring high-performance computational (HPC) resources. For instance, the Sentinel-2 satellite [11] provides approximately 200-300 GB, 3 TB, and 36 TB of daily, monthly, and annual data for the territory of Armenia. Handling this amount of data is a complex task. Therefore, HPC is the right choice to improve data processing performance using distributed computing techniques. Thus, the Big EO data processing obstacle is coping with using open-source Apache Spark [12] and Dask [13] frameworks, which can process data in parallel by dividing them into chunks, processing them in a distributed way using computational clusters, and aggregating the result. Both frameworks have master-slave architecture, where slave nodes are worker nodes executing functions in parallel, and the master node is the driver or scheduler to manage them. Spark ecosystem supports many projects in data streaming, SQL analytics, and machine learning. Spark is a multi-language engine that processes and analyzes data, while Dask is a Python library. Therefore, Spark has its ecosystem APIs and memory models, while Dask uses them from the Python ecosystem. However, these frameworks have some differences and limitations in finding an optimal solution for EO data processing workflows.

Besides using HPC, the format of EO satellite images also has a crucial influence on performance. The data compression techniques can reduce storage usage and the number of I/O operations, improving processing performance. Recent studies [14, 15] show that compression methods combined with HPC can significantly enhance the performance of Big data workflows. One of the optimal satellite image formats is Cloud Optimized GeoTIFF (COG) [16], which provides essential advantages compared to traditional formats, such as NetCDF [17]. COG format provides an HTTP range request to extract a part of the data. Hence, when extracting EO data using COG, there is no need to download the entire image and then extract the area of interest as in the NetCDF format. Besides the mentioned benefit, both COG and NetCDF formats support data compression methods.

Several studies [18, 19, 20] evaluate and compare the performance of the frameworks for particular cases, such as data-intensive neuroimaging pipelines [18], different applications of molecular dynamics [20], and scientific image analytics [19]. Nevertheless, they did not consider performance-tuning techniques, such as data compression.

The main objective of the article is to investigate the efficacy of widely used distributed data processing frameworks, such as Dask and Spark, in combination with lossless data compression methods, to enhance the performance of EO data processing. The methodology involved evaluating the approach on the Armenian hybrid research computing platform, and the results obtained from the evaluation could be used by EO communities to make informed decisions about improving their data processing performance.

2. Methodology

A test-bed platform for EO data processing has been deployed to execute EO data processing functions and compare the performances in Spark and Dask. The platform is a container-based solution within the Kubernetes system, enabling evaluating and comparing

the environments' performance. It relies on the computational resources of the Armenian hybrid research computing platform [21]. Fig. 1 shows the architecture of the experimental platform.

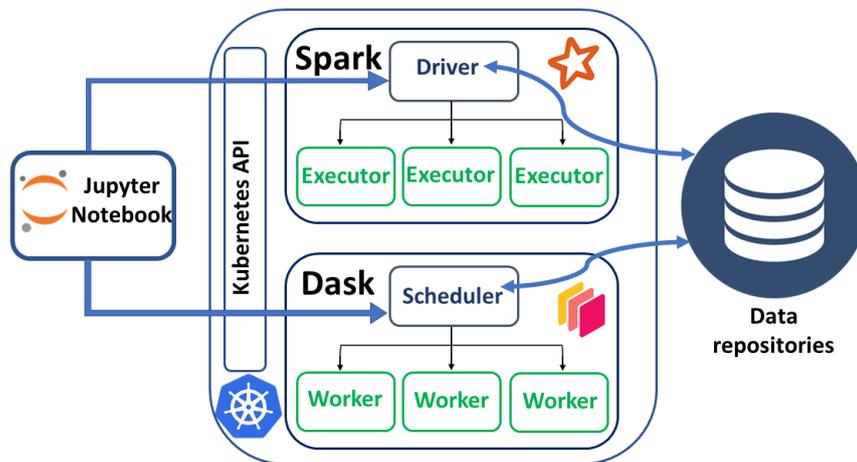


Fig. 1. Test-bed platform based on Spark and Dask.

As the figure shows, each node scheduler/driver or worker/executor corresponds to a pod in Kubernetes with some fixed computational resources. It is possible to configure the computational resource characteristics of nodes with Kubernetes API. The Jupyter Notebook [22] corresponds to the FrontEnd of the Spark and Dask cluster BackEnd. It connects to Dask and Spark of master nodes, configures environments by providing the number of worker nodes and computational resources for each node, requests to process EO data using Dask and Spark clusters, and visualizes the output. Dask and Spark clusters fetch data from repositories of either local Armenian DataCube [23] or global EO data providers. Armenian DataCube [10] provides data from Landsat 5, 7, 8 [24], and Sentinel-2 satellites, and one of the global EO data providers is Sentinel-2 Cloud-Optimized GeoTIFFs [25].

The functionality evaluation of the Dask and Spark frameworks is quite interesting. Dask is a flexible Python library, which makes it easy to migrate and execute the old-written Python code in a distributed manner. Moreover, Python is widely used in EO data workflows, and various useful libraries provide vital tools to make the work with EO data easier. However, working with EO data in Spark is a little tricky because the execution of the old-written codes in the Spark environment is impossible, as it supports APIs of its ecosystem, therefore, the code adjustment is inevitable. The GeoPySpark library [26] makes working with EO data somewhat easier in Spark. So the data processing function can be easily parallelized only in Dask, considering the limitations and complexity of using Spark.

As EO data processing applications, the Normalized Difference Vegetation Index (NDVI) [27] was evaluated during the experiments, which provides information for monitoring the health of the vegetation. The formula of the index is presented in (1).

$$NDVI = \frac{NIR - RED}{NIR + RED}, \quad (1)$$

where RED is the red band, and NIR is the near-infrared band. All bands and the calculation result are matrices or images and the NDVI index is calculated from Sentinel-2 satellite images.

Several experiments were conducted with different parameters to evaluate the performances of Dask and Spark using the developed experimental platform. Table 1 presents all parameters and their values.

Table 1: Experimental parameters and their values.

Parameter name	Possible values
Environment	Dask and Spark
Input Data sizes	16, 32, 64 GBs
Number of workers	4, 8, 16, 32
Applications	NDVI
Compression methods	None, Deflate, LZW, Packbits, and Zstandard

3. Experimental Results

Data compression techniques reduce the actual size of data, resulting in savings in storage space, providing faster network transmission times, and improving the performance of processing. EO data repositories, which provide satellite images in COG format, such as Sentinel-2 COGs, by default, use Deflate compression method to reduce the downloading time of satellite images and save some storage space. Besides the Deflate method, several compression methods, either lossy or lossless, could be applied with COGs. The accuracy of the satellite image is essential, as the spatial resolution of the Sentinel-2 image is 10m [10], which corresponds to the surface area measured on the ground represented by each pixel. Therefore, the compression methods used for optimization should be lossless to ensure accurate results. The COG format supports several lossless compression methods, such as Deflate [28], LZW [29], Packbits [30], and Zstandard [31].

EO band tiles come in three different sizes (light, medium, and heavy) by which the compression factor is estimated to understand the average compression ratio of the method. The light band tiles (coastal, water vapor, etc.) usually have up to 5-10 MB size, medium 50-70 MB (Short-wave infrared (*SWIR*), vegetation red edge, etc.), and heavy 200-250 MB (*RED*, *NIR*, etc.). They consider all types of possible lossless compression methods. The compression ratio is calculated for each method by dividing the compressed data size by the original uncompressed data size. The compression ratios for various compression methods are presented in Fig. 2.

The figure shows that the best compression factor is provided by the Zstandard method, whereas the worst one is provided by the Packbits method. Zstandard codec compresses the band image more than the Deflate does, which is by default used by the Sentinel-2 COGs repository. Therefore, using Zstandard instead of Deflate will lead to more storage savings, and less network transfer time and I/O operations. The storage reduction, in this case, is 34 % compared with the uncompressed data and 16 % compared with Deflate. The compression ratio of the Packbits method for the heavy tiles is close to 1, which means that the method is useless for data size reduction since the actual size and compressed data size will be the same. Besides the storage saving, further data processing is also essential, as

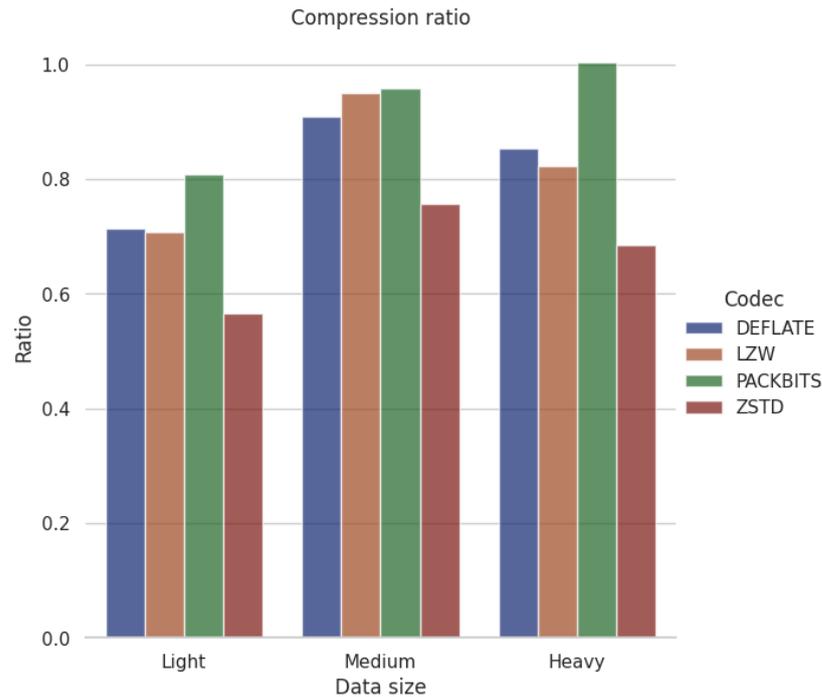


Fig. 2. Compression ratio of Deflate, LZW, Packbits, and Zstandard methods for light, medium, and heavy tiles.

high compression needs more CPU time to decompress into memory before processing. The majority of the time spent in computing NDVI is devoted to transferring satellite images over the network and loading them into memory, rather than performing calculations using the CPU. The comparison of the performances of Dask and Spark, considering different sizes of input data, compression methods, and 32 worker nodes is shown in Fig. 3.

The execution time of the COG tile compressed with the Packbits method and without compression is almost the same, as Packbits provides weak compression; thus, it uses little CPU time for decompression. The worst performance for both environments from the possible compression methods is Deflate, whereas the best one is Zstandard. Hence, the best compression method for satellite images in COG format is Zstandard, as it provides the highest compression ratio and optimal memory loading time. The performance improvement when using Zstandard compared to uncompressed mode is achieved by reducing network transfer time. Zstandard provides on average 2.15 and 1.82 times faster execution time compared with the uncompressed mode, approximately 4.72 and 3.99 times faster than the default selected Deflate method provided by global satellite image repositories correspondingly for Dask and Spark environments. Performance evaluation using Dask and Spark is quite interesting. For the default used Deflate compression method provided by EO repositories, Spark and Dask show similar execution times; however, Spark is a bit faster. The LZW compression method for the Dask environment is better than Deflate but worse than without compressing or compressing with Zstandard. Also, Spark does not support the compression method. With uncompressed data, Dask is faster than Spark for 16 GB input, whereas, in cases of 32 GB and 64 GB, Spark is faster. Performance in Dask using the Zstandard compression method is an optimal choice.

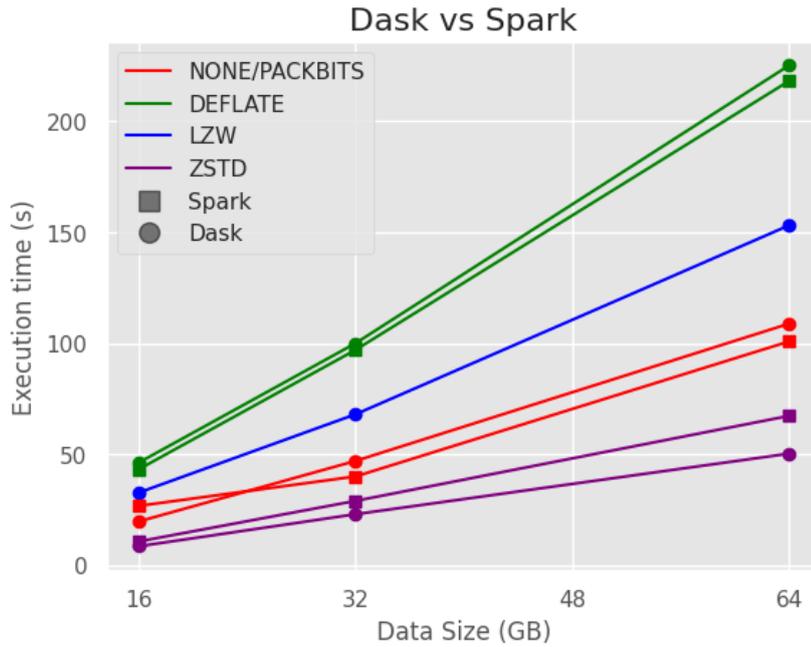


Fig. 3. Comparison of Dask and Spark considering 16, 32, 64 GBs of input data and compression methods.

4. Discussion

The study showed that various data compression methods could reduce storage requirements and network transfer time at different scales. Moreover, compressed data processing using multiple techniques in distributed environments such as Spark and Dask exhibited other execution times, with some compression methods outperforming uncompressed data processing time. The study aims to determine the optimal data compression method that balances performance and storage savings in the chosen distributed processing environments. The evaluation shows that the Dask and Zstandard combination is the best choice for the environment and compression method for EO satellite images. It provides the highest compression factor and performance compared to other supported compression methods.

The Armenian DataCube was initially set up with a 2-terabyte storage capacity, which is limited. To manage this, only the essential bands for specific EO applications that researchers are interested in during a particular period are downloaded and stored. If the storage capacity is exceeded, the options are to scale vertically or add external storage. The Zstandard compression technique was used in experiments to conserve 34 % of storage. This allows more data to be stored in the allocated DataCube space.

The Zstandard compression method combined with the Dask environment offers benefits such as improved data storage efficiency and EO data processing time. However, additional steps are required to achieve these benefits, such as converting analysis-ready data from the DataCube to Cloud Optimized GeoTIFF format and compressing them using the Zstandard method. Although this may increase the total execution time of downloading and preprocessing, it provides such benefits as enhanced processing time and storage savings. Moreover, this efficient method of storing compressed data can be applied to other types of EO data repositories and DataCubes.

In conclusion, data compression methods can effectively reduce the amount of EO data stored and improve processing performance. Zstandard exhibits the best performance and storage efficiency for EO data among the available compression methods. Additionally, the implementation of the Dask environment speeds up distributed processing.

5. Conclusion

The study evaluates the performance of EO data processing in Dask and Spark, considering compression methods. Experimental results show that Dask and Spark provide similar data processing performances. The mixture of the Dask and Zstandard compression methods is optimal, as the compression method provides the best compression factor of all possible lossless compression methods. It reduces the amount of used storage by 16 % and speeds up execution times by 4.72x and 3.99x in Dask and Spark, correspondingly compared with the Deflate method, which is used by default from the EO data repositories. In further work, it is planned to store the data in Armenian DataCube compressed with the Zstandard method and use the Dask environment for data processing.

References

- [1] O. R. Young, M. Onoda. “Satellite Earth Observations in Environmental Problem-Solving”, *In book: Satellite Earth Observations and Their Impact on Society and Policy*, pp. 3-27, 2017.
- [2] D. A. Chu, Y. J. Kaufman, “Global monitoring of air pollution over land from the Earth Observing System-Terra Moderate Resolution Imaging Spectroradiometer (MODIS)”, *Journal of Geophysical Research Atmospheres*, vol. 108, no. 21, November 2003.
- [3] R.S. dos Santos, “Estimating spatio-temporal air temperature in London (UK) using machine learning and earth observation satellite data”, *International Journal of Applied Earth Observation and Geoinformation*, vol. 88, June 2020.
- [4] T. Krishnamurti and A. Chakraborty, “Impact of Arabian Sea pollution on the Bay of Bengal winter monsoon rains”, *Journal of Geophysical Research*, vol. 114, March 2009.
- [5] R. DeFries and F. Achard, “Earth observations for estimating greenhouse gas emissions from deforestation in developing countries”, *Environmental Science & Policy*, vol. 10, no. 4, pp. 385–394, June 2007.
- [6] Y. J. Kaufman and C. Ichoku, “Fire and smoke observed from the Earth Observing System MODIS instrument–products, validation, and operational use”, *International Journal of Remote Sensing*, vol. 24, no. 8, pp. 1765–1781, November 2010.
- [7] H. D. Guo and L. Zhang, “Earth observation big data for climate change research”, *Advances in Climate Change Research*, vol. 6, no. 2, pp. 108–117, June 2015.
- [8] A. Lewis, S. Oliver and L. Lymburner, “The Australian Geoscience Data Cube Foundations and lessons learned”, *Remote Sensing of Environment*, vol. 202, pp. 276–292, 2017.
- [9] Open data cube, [Online]. Available: <https://www.opendatacube.org/>
- [10] S. Asmaryan and V. Muradyan, “Paving the Way towards an Armenian Data Cube”, *Data*, vol. 4, no. 1, 2019.

- [11] M. Drusch and U. D. Bello, “Sentinel-2: ESA’s Optical High-Resolution Mission for GMES Operational Services”, *Remote Sensing of Environment*, vol. 120, pp. 25–36, May 2012.
- [12] M. Xiangrui, “Mllib: Machine learning in apache spark”, *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1235–1241, 2016.
- [13] R. Matthew, “Dask: Parallel computation with blocked algorithms and task scheduling”, *Proceedings of the 14th python in science conference*, vol. 130, 2015.
- [14] H. Astsatryan and A. Kocharyan, “Performance Optimization System for Hadoop and Spark Frameworks”, *Cybernetics and Information Technologies*, vol. 20, no. 6, pp. 5–17, 2020.
- [15] H. Astsatryan and A. Lalayan, “Performance-efficient Recommendation and Prediction Service for Big Data frameworks focusing on Data Compression and In-memory Data Storage Indicators”, *Scalable Computing: Practice and Experience*, vol. 22, no. 4, pp. 401–412, 2021.
- [16] Cloud Optimized GeoTIFF, [Online]. Available: <https://www.cogeo.org/>
- [17] J. Li, “Parallel netCDF: A High-Performance Scientific I/O Interface”, *Proceedings of the 2003 ACM/IEEE Conference on Supercomputing*, 2003.
- [18] D. Mathieu and H. Sasson, “A Performance Comparison of Dask and Apache Spark for Data-Intensive Neuroimaging Pipelines”, *2019 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS)*, pp. 40–49, 2019.
- [19] P. Mehta and S. Dorkenwald, “Comparative evaluation of big-data systems on scientific image analytics workloads”, *Proceedings of the VLDB Endowment*, vol. 10, no. 11, pp. 1226–1237, 2017.
- [20] I. Paraskevakos and A. Luckow, “Task-parallel Analysis of Molecular Dynamics Trajectories”, *ICPP 2018: Proceedings of the 47th International Conference on Parallel Processing*, no. 49, pp. 1–10, 2018.
- [21] Y. Shoukourian and V. Sahakyan, “E-Infrastructures in Armenia: Virtual research environments”, *Ninth International Conference on Computer Science and Information Technologies Revised Selected Papers*, pp. 1–7, 2013.
- [22] B. M. Randles and I. V. Pasquetto, “Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study”, *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pp. 1–2, 2017.
- [23] Armenian DataCube, [Online]. Available: <http://datacube.sci.am/>
- [24] M. A. Wulder and T. R. Loveland, “Current status of Landsat program, science, and applications”, *Remote Sensing of Environment*, vol. 225, pp. 127–147, 2019.
- [25] Sentinel-2 Cloud-Optimized GeoTIFFs, [Online]. Available: <https://registry.opendata.aws/sentinel-2-l2a-cogs>
- [26] G. Jifu and C. Huang, “A Scalable Computing Resources System for Remote Sensing Big Data Processing Using GeoPySpark Based on Spark on K8s”, *Remote Sensing*, vol. 14, no. 3, 2022.

- [27] N. Pettorelli, J. O. Vik, “Using the satellite-derived NDVI to assess ecological responses to environmental change”, *Trends in Ecology & Evolution*, vol. 20, no. 9, pp. 503–510, 2005.
- [28] S. Oswal, A. Singh, “Deflate compression algorithm”, *International Journal of Engineering Research and General Science*, vol. 4, no. 1, 2016.
- [29] M. J. Knieser, F. G. Wolff, “A technique for high ratio LZW compression [logic test vector compression]”, *Automation and Test in Europe Conference and Exhibition*, pp. 116–121, 2003.
- [30] G. Feng, C. A. Bouman, “Efficient document rendering with enhanced run length encoding”, *Color Imaging XI: Processing, Hardcopy, and Applications*, January 2006.
- [31] Y. Collet, M. Kucherawy, “Zstandard Compression and the ‘application/zstd’ Media Type”, *RFC Editor, USA*, February 2021.

Dask-ի և Spark-ի կատարողականի վերլուծություն՝ հաշվի առնելով տվյալների սեղմումը Երկրի դիտարկման տվյալների մշակման համար

Արթուր Գ. Լալայան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ, Երևան, Հայաստան
e-mail: arthurlalayan97@gmail.com

Անփոփում

Բարձր կատարողական հաշվարկը լավ ընտրություն է երկրի դիտարկման մեծ տվյալների մշակման համար, ինչը թույլ է տալիս տվյալների մշակումը բաշխված և բարձր արդյունավետությամբ? օգտագործելով հիշողության մեջ հաշվողական հարթակներ: Տվյալների սեղմման տեխնոլոգիան նվազեցնում է պահանջվող պահեստավորման ծավալը և ցանցի փոխանցման ժամանակը, ինչպես նաև բարելավում է տվյալների մշակման ժամանակը: Հողվածի նպատակն է ուսումնասիրել լայնորեն օգտագործվող տվյալների մշակման շրջանակների արդյունավետությունը՝ տվյալների անկորուստ սեղմման տեխնիկայի հետ համատեղ, Երկրի դիտարկման հատուկ աշխատանքային հոսքերի համար սեղմման օպտիմալ մեթոդ և մշակման շրջանակ գտնելու համար: Բուսականության նորմալացված տարբերության ինդեքսը գնահատվել է Հայաստանի տարածքի համար՝ օգտագործելով Sentinel արբանյակի տվյալները և հաշվի առնելով սեղմման աջակցվող մեթոդները հիշողության մեջ Dask և Spark շրջանակների աշխատանքի համեմատման համար: Փորձերը ցույց են տալիս, որ Zstandard սեղմման մեթոդը և Dask միջավայրը լավագույն ընտրությունն են նման աշխատանքային հոսքերի համար:

Բանալի բառեր՝ Երկրի դիտարկում, HPC, Spark, Dask, բաշխված հաշվարկ, տվյալների սեղմում:

Анализ производительности Dask и Spark для обработки данных наблюдения Земли с учетом сжатия данных

Артур Г. Лалаян

Институт проблем информатики и автоматизации НАН РА, Ереван, Армения
e-mail: arthurlalayan97@gmail.com

Аннотация

Высокопроизводительные вычисления являются хорошим выбором для обработки больших данных наблюдения Земли, позволяя обрабатывать данные распределенным и высокопроизводительным способом с использованием вычислительных платформ в памяти. Технология сжатия данных сокращает объем хранилища и время передачи по сети и повышает производительность обработки. Целью статьи является исследование эффективности широко используемых систем распределенной обработки данных в сочетании с методами сжатия данных без потерь, чтобы найти оптимальный метод сжатия и структуру обработки для конкретных рабочих процессов наблюдения Земли. Нормализованный разностный индекс растительности был оценен для территории Армении с использованием данных со спутника Sentinel и с учетом поддерживаемых методов сжатия для сравнения производительности фреймворков Dask и Spark в памяти. Эксперименты показывают, что метод сжатия Zstandard и фреймворк Dask являются наилучшим выбором для таких рабочих процессов.

Ключевые слова: Наблюдение Земли, HPC, Spark, Dask, распределенные вычисления, сжатие данных.