Enhancing Earth Observation Data Processing through Optimized Multi-Modular Service

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

Hrachya Astsatryan
Institute for Informatics and Automation Problems
Yerevan, Armenia
e-mail: hrach@sci.am

Gregory Giuliani
Institute for Environmental Sciences
University of Geneva
Geneva, Switzerland
e-mail: gregory.giuliani@unige.ch

***Abstract*—The significance of earth observation data spans diverse fields and domains, driving the need for their efficient management. Nevertheless, the exponential increase in data volume brings with it new challenges that make processing and storing data more complicated. In response to these challenges, this article proposes an optimized multi-modular service for earth observation data management. The suggested approach focuses on choosing the optimal configurations for the storage and processing layers to improve the performance and cost-effectiveness of managing data. By employing the recommended optimized strategies, earth observation data can be managed more effectively, resulting in fast data processing and reduced costs.**

***Keywords*— Earth observation, distributed computing, performance optimization.**

I. Introduction

Earth observation (EO) data acquired from satellites plays a key role in various domains, including environmental monitoring [1], land cover analysis [2], water resource management [3], and global climate change studies [4]. Despite the broad utilization of EO data, the storing, management, and data processing pose significant challenges owing to its continuous expansion caused by daily observations from numerous satellites. To tackle the challenges posed by EO data, a range of technologies have been developed and implemented with the primary goal of simplifying their management. To address the complexity of storing EO data, innovative formats like Cloud Optimized GeoTIFF (COG) [5] have been proposed. This format offers significant advantages such as optimized storage in cloud environments, enabling faster access, efficient retrieval, and seamless processing of vast amounts of EO data. The format's primary advantages can be summarized in two key aspects. First, COG utilizes a tiled structure that covers square areas of the primary raster image, enabling clients to request specific data sections through HTTP range requests. Second, the format supports data compression techniques, optimizing data transfer over the internet and reducing storage utilization for more efficient handling of EO data. To overcome the challenges posed by the extensive processing of large-scale EO data, the EO community effectively employs high-performance computing (HPC) techniques. A popular choice by the EO community among these techniques is the utilization of the Dask framework [6]. The solution enables the concurrent processing of EO data by distributing the workload across multiple computational nodes, resulting in efficient and expedited data processing.

To manage EO data optimally and efficiently, it is crucial to address both storage and processing aspects.

In the data storing layer, adopting innovative solutions like COG is essential to ensure efficient EO data storage. However, it's important to note that the COG format supports various data compression methods, each of which can impact both storage savings and processing speed differently. A high compression factor can significantly reduce the data size, but it may require more time to decompress the data during processing. On the other hand, a weak compression factor may not reduce the data size and therefore the network transfer time much, thus saving less storage space, but it may result in faster processing times [7]. Finding the optimal compression method becomes a challenging task, as striking the right balance between storage savings and processing speed is essential. It requires careful consideration and testing to determine the compression method that best suits the specific requirements of handling EO data adeptly and efficiently. Storing EO data in data repositories with the most suitable compression method will result in storage savings, reduced network transfer time, and faster processing.

To ensure efficient performance in EO data processing using distributed computing, several vital factors must be taken into account. These factors encompass the cluster's configuration, determining the number of worker nodes, as well as their specific characteristics such as the number of CPUs and

https://doi.org/10.51408/csit2023_19

RAM size. Additionally, various objectives become critical for clients. Those lacking their own computational cloud infrastructure must rely on resources from global cloud providers, which come with associated costs based on the chosen options. Thus, when aiming to select an optimal cluster configuration, it becomes essential to strike a balance between multiple objectives. This entails finding the best trade-off between various factors to achieve efficient processing while considering cost, performance, and other relevant considerations. Besides this, another challenge is setting up the cluster itself, which is in addition to the effort of selecting the cluster design that is most appropriate in terms of computing complexity. As a result, a rapid and automated cloud-based provisioning and scaling solution for high-performance computing (HPC) is required.

Efficient management of EO data requires a thorough evaluation of both storage and processing layers. It is essential to make informed decisions regarding data compression and cluster configuration setup for storage and processing layers. The article presents numerous separate optimization research works and results that have recently been implemented and studied as parts of a multi-modular service, the goal of which is to provide optimization methods for efficiently handling EO data.

II. Multi-modular service

The architecture of the suggested multi-module service is shown in Figure 1.

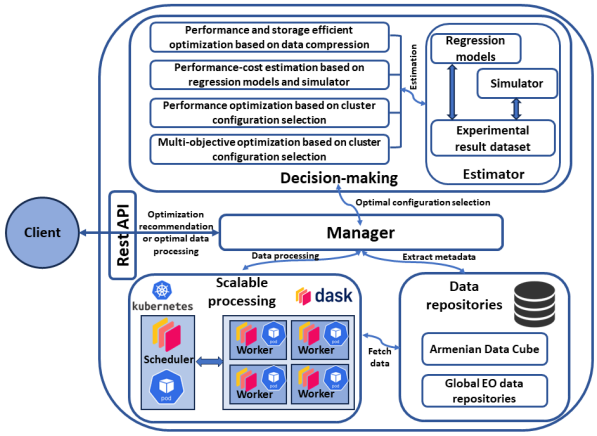

Fig. 1. Architecture of the multi-modular service

The multi-modular service consists of several modules, including Manager, Data repositories, Scalable processing, and Decision-making with the Estimator submodule. A client can access optimization methods for EO data provided by the service through the REST API.

A. Manager

The Manager module is responsible for handling the client's requests. The clients can make multiple requests to the Manager for various tasks, including:

- Recommending a data compression method for storing data in the repository,
- Estimating the execution time or the price of the computational resources required for a specific task,
- Optimization of the choice of cluster configuration, taking into account single or multiple objectives like performance and cost,
- Providing the possible optimal data processing while accounting for the mentioned optimization methods.

The Manager module is the central element that handles the service's overall functionality. Through information sharing, it works with other modules to manage and complete client requests. It cooperates with the Decision-making module to obtain the optimal configurations, works with the Data repository module to retrieve metadata about the needed data for processing, and interacts with the Scalable Computing module to process the data using the specified cluster configuration.

B. Data repositories

The EO data is stored in the Data repositories module, which also provides an API for retrieving the data. The repositories also offer another API that is intended just to deliver metadata rather than actual data. This metadata is lightweight compared with the actual data and contains essential details about the chosen region, such as the shapes of the satellite image. It is possible to determine the precise size of the processable data for the client's request by using the metadata. This method enables effective data handling without requiring the transmission of the complete dataset, conserving bandwidth and computing power. The service can calculate the data size and properly handle the client's requests with the help of the metadata.

The service is intended to handle repositories that provide either the SpatioTemporal Asset Catalog (STAC) API [8] or the Web Coverage Service (WCS) [9]. The Armenian Datacube [10], which stores EO data collected by several satellites over the area of Armenia, is now compatible with the service. The software can also communicate with other global EO data sources that offer the aforementioned kinds of APIs. Using configuration files, it is possible to configure the repositories. These files keep track of important data including the API's base URL and type (WCS or STAC). These settings allow the service to easily connect and communicate with numerous data sources, facilitating the quick retrieval and processing of EO data in response to varied customer demands.

C. Scalable processing

The EO data processing responsibilities are handled by the Scalable processing module. It uses client-requested choices such as region of interest, time, and particular bands required for the processing function to collect the necessary EO data from data sources. When it receives the required information, the module creates a Dask cluster. The Manager supplies the necessary parameters for building this cluster, including data from the Decision-making module on the ideal number of nodes and each node's computing characteristics. The specified

Dask cluster is then used by the Scalable processing module to process the data. This module guarantees quick, automated provisioning and scaling of cloud resources, enabling effective management of computing resources by processing demands. Recent implementation [11] provides automatic scalability and fast resource provisioning by using the Dask distributed package with the remote management tools deployed on a Kubernetes cluster. With this configuration, the service can effectively manage resources and scale flexibly in response to workload. A pod in the Kubernetes cluster corresponds to one worker node in the Dask cluster. According to the setup of the service, each pod is given access to particular processing resources, such as CPU and RAM. This matching of worker nodes to pods guarantees that processing activities may be divided and carried out concurrently across the available resources, making effective and parallel use of the computing capacity of the Kubernetes cluster. The service can adjust to changing workloads and processing needs by utilizing Dask automatic scaling and resource allocation features. The service may automatically spawn more pods with the right resources to tackle the workload as the quantity or complexity of processing jobs grows using the recommendation of the Decision-making module. This scaling strategy guarantees that the service can effectively analyze EO data while maximizing the usage of the computational resources capabilities of the underlying Kubernetes cluster and enables the service to handle large-scale EO data processing tasks effectively while optimizing the computational resources for faster and more responsive data processing.

D. Decision-making

The Decision-making module provides improved methods for managing EO data for both storage and processing layers. By carefully choosing the best setup, this optimization is achieved. The Estimator submodule, which includes a simulator and trained regression models constructed on historical experimental datasets, works with the Decision-making module to produce these most suitable configuration suggestions. The module offers two different sorts of recommendations: first, it specifies the optimal data compression technique to save EO data effectively for later performance-efficient distributed processing, and second, it aids in choosing the appropriate cluster for optimal distributed processing.

The results of a recent study [12] determine which data compression technique is optimal. The suggested method entails estimating the execution time of data processing while accounting for various data compression techniques and distributed computing clusters with different numbers of nodes and resources. Regression models using training data are used to make these predictions. The Decision-making module then suggests the optimal data compression technique for effective data storage based on the prediction results. EO data can be compressed using a variety of compression methods, each of which gives a unique compression ratio. As a result, the decompression duration differs between these various methods. The evaluation of the study focuses on determining how well-distributed computing environments handle data. This assessment especially takes into account the supported EO data compression techniques on various clusters, each of which has a distinct number of nodes. This examination compares the effects of the Dask and Spark environments on the speed of data processing. Study shows [13] that Dask and Spark both offer comparable data processing performance. However, combining the Dask environment with the Zstandard compression technique yields the best performance results. In comparison to all other potential lossless compression techniques, this combination produces the most beneficial compression factor. It considerably reduces execution times by around 4.72 times in Dask and 3.99 times in Spark compared to default techniques. This result demonstrates the value of combining the Zstandard compression technique with the Dask environment to produce higher data processing performance.

It is crucial to evaluate the task's execution time across a range of potential cluster configurations to choose the optimal cluster for attaining performance-efficient distributed computing of EO data. The trained regression models and a simulator included in the Estimator submodule aid in this evaluation procedure. A simulator that is particularly made for EO data processing processes has been suggested in a recent study [14]. This simulator is based on the CloudSim simulator and is utilized to estimate the execution time of EO data processing tasks. The size of the input data, which relies on the period, region, and bands, as well as the complexity of the designated function, are two criteria that are taken into consideration throughout the estimating process. The simulator also takes into account the client-described cloud infrastructure. The evaluation results demonstrate the high accuracy of the simulator in comparison to actual experiments. It is worth mentioning that the simulator obtains an R2 value of 0.88 and an RMSE (Root Mean Square Error) of 78 while forecasting the weekly Normalized Difference Vegetation Index (NDVI) for the Armenian area. Besides evaluating the execution time, the simulator can be used to determine the cost of calculation as well. The simulator considers this while running simulations because global cloud providers charge for their resources. Thus, the simulator turns into a useful tool and may be used to assess the execution time for a certain job and determine the computation cost for different kinds of clusters. Users can investigate various cluster configurations through these experiments and assess the performance and financial effects they have. The optimal cluster configuration for the particular task can be found by examining the data produced from the simulator and finding the best trade-off balance between performance efficiency and cost-effectiveness. This gives decision-makers the ability to make well-informed decisions when choosing the best cluster configuration to meet their unique processing needs while successfully controlling related expenses.

The proposed simulator and trained regression models offer methods to assess the execution time and computation cost of a task for a limited set of potential Dask clusters, which can be deployed within the client's described cloud infrastructure.

To tackle the challenge of selecting the most suitable cluster configuration, the study [15] suggests a multi-objective optimization method for optimal EO data processing, which takes both performance and cost objectives into consideration. The solution involves generating a set of possible configurations for the distributed data processing framework, which serves as a Pareto frontier. By employing this approach, the optimal cluster configuration that aligns with their specific needs and constraints can be used, ensuring an efficient balance between data processing performance and computation cost for EO data processing tasks. The evaluation of the experiments shows that the performance can rise by as much as 1.66 times while costs can decrease by a factor of 2.38 in some scenarios using the suggested method.

III. Conclussion

The paper proposed a multi-modular service for enhancing earth observation data processing that combines numerous separate optimization research investigations and studies. This service's major goal is to provide the optimal configuration selection for efficiently handling EO data at both the storage and processing layers. The goal is achieved by enabling well-informed choices for cluster architectures and data compression algorithms. Further activities: it is planned to increase the estimating module's accuracy while lowering error rates in the optimization module, implement OLAP-based services, and several data processing functions to help the earth observation community properly monitor the environment.

Acknowledgment

The research was supported by the Science Committee of the Republic of Armenia by the project entitled "Scalable data processing platform for EO data repositories" (Nr. 22AA-1B015) 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".

References

[1] G. Giuliani, E. Egger, J. Italiano, C. Poussin, J.-P. Richard and B. Chatenoux, "Essential Variables for Environmental Monitoring: What Are the Possible Contributions of Earth Observation Data Cubes?", *Data 2020*, vol. 5, no. 4, 2020.

[2] S. K. Singh, P. B. Laari, S. Mustak, P. K. Srivastava and S. Szabó, "Modelling of land use land cover change using earth observation data-sets of Tons River Basin, Madhya Pradesh, India", *Geocarto International*, vol. 33, no. 11, 2020.

[3] R. Guzinski, S. Kass, S. Huber, P. Bauer-Gottwein, I. H. Jensen, V. Naeimi, M. Doubkova, A. Walli and C. Tottrup, " Enabling the Use of Earth Observation Data for Integrated Water Resource Management in Africa with the Water Observation and Information System", *Remote Sensing*, vol. 6, no. 8, 2014.

[4] H. D. Guo, L. Zhang and L. W. Zhu, " Earth observation big data for climate change research", *Advances in Climate Change Research*, vol. 6, no. 2, pp. 108–117, 2015.

[5] Cloud Optimized GeoTIFF. [Online]. Available: https://www.cogeo.org

[6] M. Rocklin, "Dask: Parallel computation with blocked algorithms and task scheduling", *Proceedings of the 14th python in science conference*, SciPy Austin, TX, vol. 130, 2015.

[7] H. Astsatryan, A. Kocharyan, D. Hagimont, and A. Lalayan, "Performance Optimization System for Hadoop and Spark Frameworks", *Cybernetics and Information Technologies*, vol. 20, no. 6, pp. 5–17, 2020.

[8] SpatioTemporal Asset Catalogs. [Online]. Available: https://stacspec.org/en

[9] P. Baumann, "Beyond rasters: introducing the new OGC web coverage service 2.0", *In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 320–329, 2010.

[10] S. Asmaryan, V. Muradyan, G. Tepanosyan, A. Hovsepyan, A. Saghatelyan, H. Astsatryan, H. Grigoryan, R. Abrahamyan, Y. Guigoz, G. Giuliani, "Paving the Way towards an Armenian Data Cube", *Data*, vol. 4, no. 3, 2019.

[11] H. Astsatryan, A. Lalayan and G. Giuliani, "Scalable Data Processing Platform for Earth Observation Data Repositories", *Scalable Computing: Practice and Experience*, vol. 24, no. 1, pp. 35–44, 2023.

[12] H. Astsatryan, A. Lalayan, A. Kocharyan and D. Hagimont, "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.

[13] A. Lalayan, "Data Compression-Aware Performance Analysis of Dask and Spark for Earth Observation Data Processing", *Mathematical Problems of Computer Science*, vol. 59, pp. 35–44, 2023.

[14] A. Lalayan, H. Astsatryan and G. Giuliani, "Earth Observation Data Processing Simulator Based on the CloudSim", *The 14th International Conference on Large-Scale Scientific Computations*, Sozopol, Bulgaria, 2023.

[15] A. Lalayan, H. Astsatryan and G. Giuliani, "A Multi-Objective Optimization Algorithm for EO Data Processing Based on Dask Library", *Proceedings of the 13th Conference "Data analysis methods for software systems"*, Druskininkai, Lithuania, 2022.