On the Realisation of a Workflow for Continuous Earth Observation of Forest Dynamics: A Performance Engineering Challenge

Nikolas Herbst, Niklas Jaggy, David Dingel, David Linke, Claudia Kuenzer, Samuel Kounev
Julius-Maximilians-University, Würzburg, Germany

Abstract

Up-to-date data on forest dynamics is vital for forest management and understanding their impact on biodiversity and climate change mitigation. In this context, remote sensing has emerged as promising solution, especially for large-scale forest-related scenarios. We are implementing a satellite data processing workflow to continuously feed a mobile application: The goal is to provide timely and targeted information on forest dynamics, local disturbance events, and biodiversity changes for the entire Bavaria region.

The engineering, automating and scaling of such a data-intense and distributed processing workflow, from high-volume satellite data time series to mobile applications, poses a variety of performance engineering challenges. We showcase first measurements of individual workflow task: The resource demanding disturbance detection executed in a Python Dask HPC environment contrasted against the resource-bound, optimized DBSCAN clustering of approximately 22 million points into (currently) multiple 100k disturbance events in the mobile application back-end.

1 Introduction and Context

In the context of accelerated climate change, the provision of digital information on forest dynamics plays a pivotal role in climate-resilient forest management and sustainable forestry. They enable decision-makers at all levels to make evidence-based decisions and act promptly. Germany, by nature, is predominantly covered with forests, yet human activity has reduced forest coverage in Bavaria to approximately one-third in favor of agriculture, settlement, and transportation. Nevertheless, the remaining forested area serves numerous essential ecosystem functions for society [7].

The development of efficient approaches to process Earth observation data requires not only scalable computing power but also the ability to deal with large amounts of data (up to petabytes) from distributed and heterogeneous data sources [5]. Geoscientists are often faced with a range of challenges, including the large number of possible algorithms, computation and I/O-intensive processes, and the lack of standards for workflow specification and data description [2, 4].

High-resolution satellite data from the European Sentinel missions now offer extensive opportunities

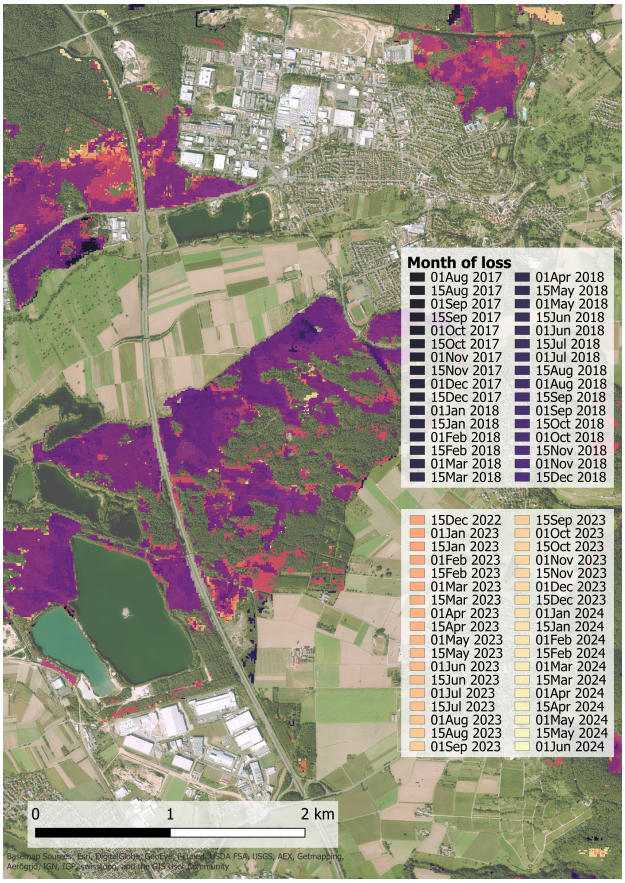

Figure 1: A local example of biweekly forest disturbance detection in north-western Bavaria.

to capture and quantify various changes in forests in great detail, as demonstrated in numerous local studies [1, 3, 8]. What is currently lacking, however, are up-to-date, comprehensive geoinformation products (e.g., covering entire states) and the continuous provision of this information (i.e., multiple times per month). Only then can this information contribute to improved management decisions.

In the ongoing ROOT project[1], we are developing and establishing earth observation (EO) workflows based on prior work to provide at least bi-weekly information on stand losses in the entire Bavarian state, both for the past five and upcoming years. This infor-

[1]ROOT project page: `https://se.informatik.uni-wuerzburg.de/root`

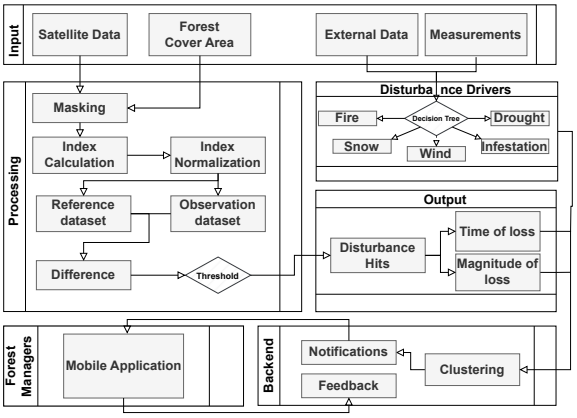

Figure 2: ROOT workflow from satellite data processing to presenting results to forest managers via a mobile app.

mation is then quantified in graphics, differentiated by clear-cuts and standing deadwood, and categorized for the causes of stand changes (drying out, pests, wind throw, fire, timber harvest, etc.). It will be made available through a geoinformation portal, a smartphone app, and reports.

From a technical perspective, we are addressing three core activities for the developed processing workflow and information services: (1) automation of processing steps for various geoinformation products, (2) optimization of performance, energy efficiency, and scalability of data processing, (3) preparation and continuous provision of results in high-resolution through a smartphone app. We present the high-level design of our ROOT approach in Figure 2. The workflow starts with data input streams into continuous processing of disturbance masks, assigning found disturbance drivers to individual (clustered) disturbances in a managing backend. The backend then generates notifications on new or changed disturbances for registered app-users in their respective subscribed areas and accumulates feedback from them on the actual situation on-site.

2 Initial Performance Measurements

2.1 EO Time Series Processing in HPC Environment

The EO processing chain is implemented in a modular way to separate the EO-based processing steps. Consequently, the data preprocessing, data preparation and disturbance detection stages are self-contained objects. Due to the high resolution (10m per pixel), size of the study area (entire Bavarian state) and dense time series across more than 6 years to process, we relied on the Dask open-source python library that allows efficient parallel processing of large multi-dimensional array data [10]. This made it possible to scale the processing to a distributed cluster on the terrabyte HPDA [6, 9, 11] and proved to be necessary to transform the 4D input satellite data into a meaningful 3D spectral index time series.

In order to assess the efficiency of the processing chain we compare the processing statistics of several 35x35 km tiles with varying amount of forest cover from the grid used to cover entire Bavaria (Table 1). This revealed how the Dask cluster performs on different input data sizes. Additionally, we investigated how the cluster behaves for repeated runs of the same tile. For all test runs the setup of the Dask cluster was the same with 15 CPU's distributed as 15 workers with 40 GB RAM for each worker.

Tile	Runtime	CPU	maxRam
Tile 1 42.96 km^2	25m32s	1103%	77.5gb
Tile 2 669.82 km^2	Run 1: 2h44m Run 2: 2h40m	1535%	381.8gb 372.4gb
Tile 3 410.68 km^2	Run 1: 1h18m41s Run 2: 1h43m10s	1477%	381.9gb 360.6gb

Table 1: Comparison of tile processing performance.

2.2 DBSCAN-based Clustering in Compact VM

The Disturbance Management Component is a Quarkus (Java-based) mobile application back-end and receives the updated disturbance masks from the processing. The disturbed pixels are then clustered into disturbance events using a tailored DBSCAN implementation executed on a compact VM with 4GB memory. The task-size is the same for all runs: The mask contains about 22 million pixels that are clustered into mutiple 100k clusters. Depending on execution mode, we observe significantly different performance behavior (Fig. 3): 1) Sequential, plain Java streams, persist phase in the second half; 2) Concurrent streams merging and persisting phase; 3) Piped, 8 worker threads in a pool for clustering, one for merging, back-pressure controlled concurrency level and persisting.

3 Conclusion and Next Steps

Having implemented the first end-to-end prototype, we are now thoroughly analyzing performance, scalability, efficiency, and real-time capability of the workflow tasks. We plan to apply the Common Workflow Language CWL[2] formalism (capturing the workflow task dependencies in a directed acyclic graph DAG) and extend it by resource demand annotations depending on data input characteristics. Towards continuous operation, we require caching and reusing intermediate results to optimize performance and resource efficiency. On a technical level, we plan to deploy suitable tasks in a serverless environment.

[2]CWL: `https://www.commonwl.org/`

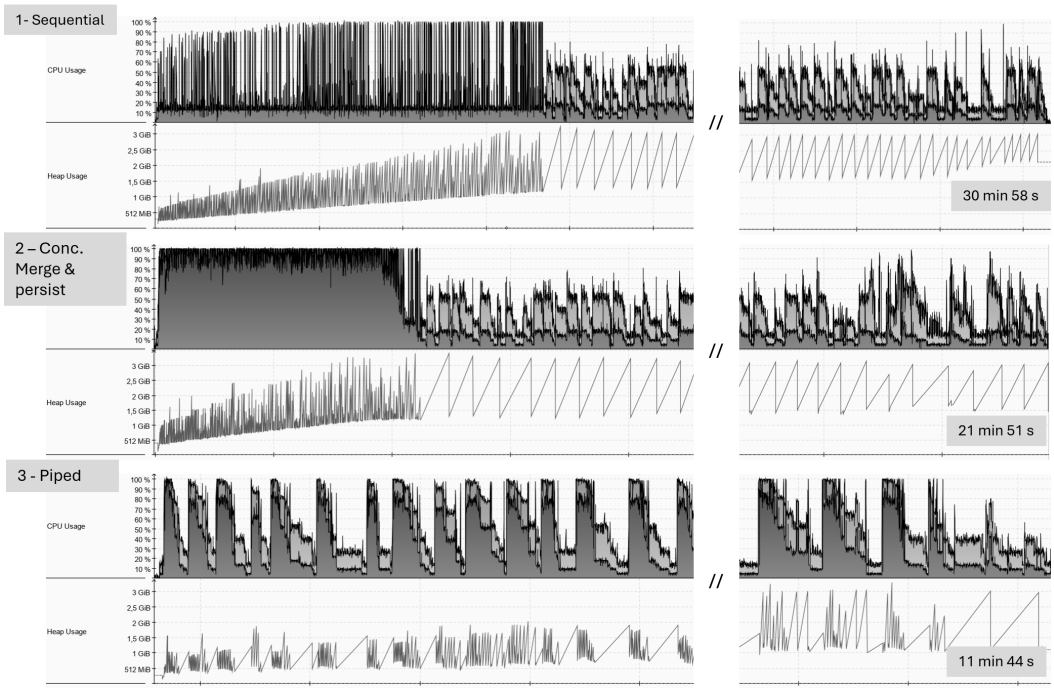

Figure 3: DBSCAN-based clustering of a disturbance mask in 3 execution modes.

4 Acknowledgement

This research project is funded by the Bavarian Research Institute for Digital Transformation (bidt), an institute of the Bavarian Academy of Sciences and Humanities. We thank our students that support this work, namely David Linke, David Dingel, Philipp Klein and Martin Klüpfel.

References

[1] J. Müller and R. Brandl. "Assessing biodiversity by remote sensing in mountainous terrain: the potential of LiDAR to predict forest beetle assemblages". In: *Journal of Applied Ecology* 46.4 (July 2009), pp. 897–905. DOI: 10.1111/j.1365-2664.2009.01677.x.

[2] N. E. Young et al. "A survival guide to Landsat preprocessing". In: *Ecology* 98.4 (Mar. 2017), pp. 920–932. DOI: 10.1002/ecy.1730.

[3] S. Bae et al. "Radar vision in the mapping of forest biodiversity from space". In: *Nature Communications* 10.1 (Oct. 2019). DOI: 10.1038/s41467-019-12737-x.

[4] C. Yang et al. "Big Earth data analytics: a survey". In: *Big Earth Data* 3.2 (Apr. 2019), pp. 83–107. DOI: 10.1080/20964471.2019.1611175.

[5] V. Gomes, G. Queiroz, and K. Ferreira. "An Overview of Platforms for Big Earth Observation Data Management and Analysis". In: *Remote Sensing* 12.8 (Apr. 2020), p. 1253. DOI: 10.3390/rs12081253.

[6] J. Eberle, M. Schwinger, and H. Zwenzner. "Towards Scientific and Interoperable Earth Observation Exploitation Platforms". In: *Proceedings of the 2021 Conference on Big Data from Space (BiDS'21)*. Bukarest, Rumania: Publications Office of the European Union, 2021, pp. 89–92. DOI: 10.2760/125905.

[7] N. K. Simons et al. "National Forest Inventories capture the multifunctionality of managed forests in Germany". In: *Forest Ecosystems* 8.1 (Jan. 2021). DOI: 10.1186/s40663-021-00280-5.

[8] J. Koehler et al. "Towards forecasting future Snow Cover Dynamics in the European Alps – The Potential of Long Optical Remote-Sensing Time Series". In: (July 2022). DOI: 10.20944/preprints202207.0290.v1.

[9] J. Eberle, M. Schwinger, and J. Zeidler. "Challenges in the development of the EO Exploitation Platform terrabyte". In: *Proc. of the 2023 Conf. on Big Data from Space (BiDS'23) – From Foresight to Impact*. Vienna, Austria: Publications Office of the EU, 2023, pp. 97–100. DOI: 10.2760/46796.

[10] *Dask: A flexible open-source Python library for parallel computing*. https://www.dask.org/. Accessed: 2024-04-05.

[11] DLR. *terrabyte High Performance Data Analytics (HPDA) platform*. https://www.dlr.de/eoc/terrabyte. Accessed: 2024-04-05.