Project Application - Manuel Campagnolo

The Project

Project details

Project title: Using HPC for vegetation loss monitoring from Sentinel-2 imagery over Portugal

Project summary (abstract):

We collaborate with the Portuguese Agency for Territorial Development (DGT) towards the goal of monitoring vegetation losses over Portugal using satellite imagery. To that end, we firstly explore the full 10-20 m Sentinel-2 archive for Portugal available from 2017 and perform a comprehensive time series analysis at pixel level leading to feature-preserving dimensionality reduction of the whole Sentinel-2 image collection for change detection purposes (TASK1). Secondly, we perform a spatio-temporal classification to identify the approximate date and locations of vegetation losses with a minimum unit area of 0.5 ha (TASK2). The ultimate goal of this proposal is to develop an HPC operational pipeline and optimized code that would have been tested and validated and that DGT can put into production to provide a near-real time (bimonthly) vegetation loss product. The product will be open and made available to the community through the National Archive of Geographical Data at SNIG/DGT. In addition, the set of potential change points and their features from TASK1 will be also made available to the community. That reduced data set will be relevant for a range of environmental monitoring tasks like burned area mapping or urban development monitoring for the whole Portuguese territory that cannot in practice rely on the full Sentinel-2 collection.

Explain the scientific case of the project for which you intend to use the code(s):

This application follows an ongoing exploratory project with MACC-Deucalion (FCT 2024.10004.CPCA.A1) to benchmark our processing pipeline. Our current computational experiments indicate which resources are needed to complete processing for a complete coverage of the Portuguese territory and point to technical issues that still need to be overcome. The ongoing project 2024.10004.CPCA.A1 is being used for testing and bench-marking over a limited region, with the current application necessary to scale this pipeline for a complete coverage of Portugal.

The scientific case that motivates our work rests on two complementary methodological advances that are described below.

In Task 1, we analyze approximately 500 million pixel time-series from Sentinel-2 satellite imagery using continuous change detection techniques. By modeling temporal patterns with harmonic functions, we can identify subtle shifts in vegetation condition and estimate the timing of disturbances. This pixel-level analysis is crucial for detecting changes that more traditional remote sensing approaches based on monthly composites fail to identify.

In Task 2, we leverage these pixel-level results to identify coherent spatial patterns of change. By integrating temporal, spectral and spatial dimensions through either neural network segmentation or graph-based community detection, we can identify meaningful patches of change rather than isolated pixels, witch is consistent with the minimum 0.5 ha monitoring target of DGT.

Early detection of vegetation loss is very relevant for climate change mitigation, biodiversity conservation, and natural resource management. By providing timely, accurate information on forest disturbances, we enable rapid response that can be relevant to wildfires, illegal logging, disease outbreaks, and other threats to forest ecosystems. This directly supports Portugal's commitments to the Paris Climate Agreement and the EU Biodiversity Strategy.

By leveraging HPC resources, we transform what would be a computationally prohibitive analysis into an operational tool, bridging the gap between Earth observation science and practical environmental management.

Our previous experience and scientific publications that are the most relevant for this application are de following.
1. As mentioned earlier, this application is a follow up of a current project with MACC/Deucalion where we are currently benchmarking TASK 1 of our proposed pipeline
2. We have authored two papers that use the core algorithm PyCCD for change detection:
* D Moraes, B Barbosa, H Costa, F D Moreira, P Benevides, M Caetano, M Campagnolo. Continuous forest loss monitoring in a dynamic landscape of Central Portugal with Sentinel-2 data. International Journal of Applied Earth Observation and Geoinformation. 2024. https://doi.org/10.1016/j.jag.2024.103913
* A Alves, D Moraes, B Barbosa, H Costa, F D Moreira, P Benevides, M Caetano, M Campagnolo. Exploring Spectral Data, Change Detection Information and Trajectories for Land Cover Monitoring over a Fire-Prone Area of Portugal. GISTAM. 2023. https://doi.org/10.5220/0011993100003473
3. For TASK2, we have experience both in semantic segmentation and graph segmentation applied to remote sensing data:
* D Moraes, M Campagnolo, M Caetano. A Weakly Supervised and Self-Supervised Learning Approach for Semantic Segmentation of Land Cover in Satellite Images with National Forest Inventory Data. Remote Sensing. 2025. https://doi.org/10.3390/rs17040711

* Campagnolo, M. L., Oom, D., Padilla, M., & Pereira, J. M. C. (2019). A patch-based algorithm for global and daily burned area mapping. Remote Sensing of Environment, 232, 111288. https://doi.org/10.1016/j.rse.2019.111288 imate

Keywords:

Sentinel-2, change detection, image segmentation, I/O, MPI, CUDA

Instructions: Not provided

Proposal for civilian purposes: true

Is any part of the project confidential?: No

Research fields #1

Research field title: PE10 Earth System Science

Research field share (%): 100

AI set of technologies selection: Deep Learning, Decision management: Classified and statistical learning methods

Please specify how does your project ensure ethical principles and addresses potential societal impacts associated with the development and deployment of AI technologies:

Our satellite-based vegetation loss monitoring project for Portugal adheres to the following ethical principles:
1. We use publicly available Sentinel-2 satellite imagery available through the Copernicus Open Access Hub
2. No personally identifiable information is collected or processed
3. All derived products will be properly documented with metadata regarding methodology and limitations
4. Uncertainty in change detection results will be quantified and transparently communicated
5. Our monitoring approach prioritizes equal coverage across all regions of Portugal
6. Results will be made accessible to all stakeholders since the final results of the project will be made available to the public
7. All AI/ML algorithms used for change detection will be documented and their limitations clearly stated
8. Source code will be made available through open-source repositories to enable reproducibility and scrutiny
9. We will develop interpretive guidelines for our results
10. Regular validation against ground truth data will be performed and reported
11. Focus will remain on broad environmental patterns rather than individual land management practices
12. Knowledge transfer is a core component of our project
13. We will develop simplified tools that can be maintained locally after the project concludes
14. Documentation will be available in Portuguese and on-line to ensure accessibility

Submission details

Project duration: 12 months

Preferable starting date: 01-04-2025 (CET)

Industry involvement: None

Public sector involvement: As Team Member

Principal Investigator

Personal information

Gender: Male

Title: Prof.

First (given) name: Manuel

Last (family) name: Campagnolo

Initials: L.

Date of birth: 08-11-1965 (CET)

E-mail address: mlc@isa.ulisboa.pt

Secondary e-mail address: mlc@edu.ulisboa.pt

Nationality: Portuguese

Phone Number: +351|213653470

Job title: Associate Professor

Employment contract valid for more than 3 months after end allocation: true

Website: https://www.isa.ulisboa.pt/id/mlc; https://orcid.org/0000-0002-9634-3061

Organization details

Instructions: Not provided

Organization name: Instituto Superior de Agronomia, Universidade de Lisboa

Organization type: University

Organization with research activity: Yes

Organization head office is located in Europe: Yes

Percentage of R&D in Europe vs total R&D: 95% (estimate)

Organization department: Department of Biosystems Science and Engineering

Organization group: Mathematics

Organization address: Instituto Superior de Agronomia, Tapada da Ajuda, Lisboa

Organization postal code: 1349-017

Organization city: Lisboa

Organization country: Portugal

Contact Person and Team Members Information

Contact Person

First (given) name: Manuel

Last (family) name: Campagnolo

E-mail address: mlc@isa.ulisboa.pt

divider: Not provided

Team Members #1

Personal Information

Gender: Female

Title: Ms.

First (given) name: Sara

Last (family) name: Caetano

Initials: Not provided

Team member is a Co-PI: No

Date of birth: 17-09-1997 (CET)

E-mail address: fc48067@alunos.fc.ul.pt

Secondary e-mail address: sara.a.ctn1997@gmail.com

Nationality: Portuguese

Phone number: +351|915550748

Job title: Code developer

Website: https://www.linkedin.com/in/sara-caetano-49b026168/

Organization details

Instructions: Not provided

Organization name: FCUL (Faculty Of Sciences - University of Lisbon)

Organization type: University

Organization department: DEGGE (Departamento Engenharia Geográfica, Geofísica e Energia)

Organization group: Not provided

Organization address: Campo Grande 016

Organization postal code: 1749-016 Lisboa

Organisation city: Lisboa

Organization country: Portugal

Team Members #2

Personal Information

Gender: Male

Title: Mr.

First (given) name: Daniel

Last (family) name: Moraes

Initials: Not provided

Team member is a Co-PI: Yes

Date of birth: 06-12-1990 (CET)

E-mail address: g20180450@novaims.unl.pt

Secondary e-mail address: moraesd90@gmail.com

Nationality: Portuguese

Phone number: +351|915722789

Job title: PhD student

Website: https://danielm09.github.io

Organization details

Instructions: Not provided

Organization name: DGT - Direção Geral do Território (General Directorate of the Territory)

Organization type: Public administration or public institution

Organization department: Geospatial Intelligence Multidisciplinary Team

Organization group: Not provided

Organization address: Rua Artilharia Um, 107

Organization postal code: 1099-052

Organisation city: Lisboa

Organization country: Portugal

Team Members #3

Personal Information

Gender: Male

Title: Mr.

First (given) name: Dominic

Last (family) name: Welsh

Initials: Not provided

Team member is a Co-PI: No

Date of birth: 28-09-1991 (CET)

E-mail address: djwelsh@edu.ulisboa.pt

Secondary e-mail address: domwelsh@gmail.com

Nationality: British

Phone number: +351|965878860

Job title: MSc Student

Website: https://www.linkedin.com/in/domwelsh/

Organization details

Instructions: Not provided

Organization name: Insituto Superior de Agronomia, Universidade de Lisboa

Organization type: University

Organization department: Not provided

Organization group: Not provided

Organization address: Instituto Superior de Agronomia, Tapada da Ajuda

Organization postal code: 1349-017

Organisation city: Lisboa

Organization country: Portugal

Partitions

Partitions #1

Partition name:

Deucalion CPU X86

Code(s) used: pyccd, MPI4py, scikit-learn, h5py, SciPy, NumPy

Requested amount of resources (node hours): 3000

Average number of processes/threads: 1024

Average job memory (total usage over all nodes in GB): 2048

Maximum amount of memory per process/thread (MB): 2000

Total amount of data to transfer to/from (GB): 5000

Justification of data transfer:

Our input data consists in 500*10^6 pixels, each one representing a time series of Sentinel-2 images (from 2017 to the present). Those pixels are stored in a hdf5 file with approximately 2800 GB in size, which will have to be uploaded to the HPC machine. After processing those pixels, we need to download the intermediate and final outputs, which would amount to a total of 2200 GB in a conservative estimate. Therefore, considering download and upload we estimate that data transfer will amount to approximately 5000 GB.

Describe the I/O strategy regarding the parameters indicated below.: Not provided

Is I/O expected to be a bottleneck?:

Yes.

I/O libraries, MPI I/O, netcdf, HDF5 or other approaches:

In our initial tests with a small sample of the data, we were using the NPY file format to store our inputs. However, since the NPY format does not support compression, thus leading to very large files, we considered this approach unfeasible given the enormous volume of data to be stored and transferred. Now we are working with our inputs in the HDF5 format (with h5py), which will need to be accessed in parallel by MPI-IO (read only) for coordinated file access across nodes

Frequency and size of data output and input:

In the initial phase of the project, we are attempting to process all 500*10^6 pixels. This requires an initial upload of all pixels in multiple HDF5 files, each corresponding to a Sentinel-2 tile. We estimate that processing all pixels takes 2-3 days. In a second phase of the project, we will also analyze how to update the time series as new Sentinel-2 observations are ingested. This will require additional uploads and runs. Gains in efficiency are possible by re-using the outputs of the pixel-level time series analysis and just updating the most recent time segments.

Number of files and size of each file in a typical production run:

We envision that a typical production run's input consists in 17 HDF5 files, each representing a Sentinel-2 tile and with size ranging from 30 GB to 240 GB depending on the coverage of each tile. The time series processing will output for each pixel a collection of temporal segments, each one described by a set of features. The outputs for all pixels in the tile will be stored in a PARQUET file. Full time series information (less than 1000 images with 4 spectral bands) will be converted into a collection of harmonic time segments for each pixel that are stored into ~100 variables per segment, with in average 2 segments per pixel. The output will be exported, and will also be used for the subsequent processing in the GPU partition. This will be repeated for the 17 Sentinel-2 tiles that cover Continental Portugal.

In short,
- Input: 17 HDF5 files ranging from ~30 GB to 240 GB, totaling 2800 GB
- Output: 17 PARQUET files, totaling ~500 GB
- Intermediate files: 1700 GB

Total storage required (GB): 5000

Partitions #2

Partition name:

Deucalion GPU

Code(s) used: CUDA, Dask, DALI, PyTorch

Requested amount of resources (node hours): 400

Average number of processes/threads: 136

Average job memory (total usage over all nodes in GB): 800

Maximum amount of memory per process/thread (MB): 40000

Total amount of data to transfer to/from (GB): 1500

Justification of data transfer:

The entire 500GB parquet dataset must be loaded and processed for deep learning image segmentation. Additional transfers include intermediate processed data (500GB) and storage for model checkpoints and final segmentation results (500GB), requiring a total of 1500GB data transfer.

Describe the I/O strategy regarding the parameters indicated below.: Not provided

Is I/O expected to be a bottleneck?:

Yes, initial data loading of the large PARQUET file could be a bottleneck.

I/O libraries, MPI I/O, netcdf, HDF5 or other approaches:

PARQUET via pyarrow/Dask for input
HDF5 for intermediate processed data
NVIDIA DALI for optimized data loading pipeline
MPI for inter-node communication

Frequency and size of data output and input:

Input: One-time load of 500GB PARQUET file, streamed in chunks
Output: Model checkpoints (~5GB) every epoch (approximately 20-30 epochs)
Final output: ~100GB of segmentation masks

Number of files and size of each file in a typical production run:

Input: 1 PARQUET file (500GB)
Intermediate: ~100 HDF5 files (~5GB each)
Output: ~1000 files (100MB each) for segmentation results
Checkpoints: ~30 files (5GB each)

Total storage required (GB): 1500

Code Details and Development

Instructions: Not provided

Development of the code(s) description:

The code is available at https://github.com/manuelcampagnolo/S2CHANGE/tree/main/scripts/

The code for TASK1 is written in Python and has essentially two parts. (1) A local module that converts input satellite data into npy or hdf5 format for N pixels. (2) A module to be run on HPC that parallelizes processing for the N time series that correspond to those pixels. Each pixel is processed independently using module PyCCD. For the totality of the Sentinel-2 tiles we process, N is approximately 500 million.

Function "main_mpi.py". Input: either npy or hdf5 file that contains time-series values of pixels to be processed. Output: PARQUET file, where each row corresponds to a segment from the time series analysis of a single pixel. Each time series is processed with "pyccd.py". The processing is independent for all pixels within the input mask. Each time series corresponds to all values of the Sentinel-2 collection for one given pixel. This is the function that evenly distributes a batch of pixels by nodes and threads.

Function "pyccd.py". Input: temporal series with approximately 1000 observations; Output: a list of one or more time segments, each one described by a set of ~100 variables

The code for the classification task (TASK2) to be executed in the GPU partition is not yet available. The code for Task 2 aims at creating groups (patches) of pixels from the features extracted from the output of the time series analysis in TASK 1 and the geographical location of those pixels.

The inputs and outputs for TASK 2 are the following.

Input: PARQUET file where each row is a list of features of a time segment. Each segment has a change date and a probability of change. There can be more than one time segment per pixel. This file includes information about all Sentinel-2 pixel s over Portugal where potentially vegetation loss can occur (500 million pixels). Additional input for training and test data.

Output. Indices of pixels that form spatial patches of pixels that have a similar change date. The final output is a spatio-temporal map that indicate where and when changes occurred. Validation results.

The algorithm for grouping pixels in patches of predicted change can be either (1) a semantic segmentation neural network as in (Moraes et al, 2025) or (2) a graph based method where groups are identified with a fast graph based technique like maximum modularity as in (Campagnolo et al, 2019). Recently, CUDA implementation of the Louvain algorithm for maximum modularity have been proposed via the RAPIDS ecosystem library.

Code details #1

Name and version of the code:

ccdISA, version 1

Webpage and other references:

Current development: https://github.com/manuelcampagnolo/S2CHANGE/tree/main/scripts

Original pyccd code: https://github.com/repository-preservation/lcmap-pyccd

Licensing model:

Current development: MIT license

Original pyccd code: Unlicense license

Contact information of the code developers:

Manuel Campagnolo: mlc@isa.ulisboa.pt
Daniel Moraes: g20180450@novaims.unl.pt
Sara Caetano: fc48067@alunos.fc.ul.pt
Dominic Welsh: djwelsh@edu.ulisboa.pt

Your connection to the code (e.g. developer, collaborator to main developers, etc.):

coordinator and collaborator

Scalability and performance

Describe the scalability of the application and performance of the application:

TASK1. The application has excellent scalability due to the embarrassingly parallelism of pixel-wise time series analysis. Each pixel's processing is completely independent, allowing for efficient distribution across multiple nodes and threads with minimal communication overhead. The code uses MPI for distribution of workloads, with each node processing batches of pixels simultaneously. Current performance scales nearly linearly with the addition of compute resources up to hundreds of cores. With the current implementation, and according to early tests, 250 threads process 10^6 pixels is less than 10 minutes, with the full HPC deployment expected to process all 500 million pixels in under 24 hours.

TASK2. Using GPU acceleration, it is clear that the neural network approach is adequate for semantic segmentation. The graph-based method demonstrates different scaling characteristics, with performance primarily bounded by memory capacity rather than compute.

What is the target for scalability and performance?:

TASK1. Our target is to process all 500 million pixels within a 24-hour computation window. This performance target is essential to enable rapid iteration of our change detection algorithms, allowing multiple experimental runs within the project timeline. Additionally, achieving this performance would make it feasible to update vegetation loss maps on a bimonthly basis as new Sentinel-2 data becomes available, greatly enhancing the scientific value by providing near real-time monitoring capabilities for land management decisions.

TASK2. For training and validation a subset of the original data set should be processed within less than one hour to facilitate the development of the algorithm. For production, our scientific goals require processing the entire 500 million pixel dataset and updating the output maps every two months. However, testing should be part of the production pipeline for quality control, and therefore, our target of 24 hours would permit adjustments and algorithm fine-tuning to guarantee a sustained level of precision.

Optimization of the work proposed

Explain how the optimization work proposed will contribute to future Tier-0 projects:

TASK1. Sentinel-2 is available globally and in particular for the EU. So efficient HPC code can be replicated for other regions of Southern Europe with similar characteristics in terms of data availability and cloud cover. The optimizations developed in this project will establish a framework for continental-scale land cover change detection that can be deployed on Tier-0 systems, enabling improved temporal and spatial resolution in Earth observation analytics.

TASK2. The optimization work developed for this task will establish scalable methods for GPU-accelerated spatio-temporal analysis of Earth observation data.

Describe the impact of the optimization work proposed - is the code widely used; can it be used for other research projects and in other research fields with minor modifications; would these modifications be easy to add to the main release of the software?:

TASK1. The PyCCD code is already widely used within the remote sensing community for land cover change detection. Our optimized HPC implementation will be published as an open-source extension to the existing codebase, making it immediately accessible to hundreds of research groups. The modular nature of our implementation allows for easy adaptation to different sensors (Landsat, Sentinel-1, PlanetScope) with minimal code changes. The time series analysis methods can be readily applied to other environmental monitoring applications, namely burned area mapping.

TASK2. The code will be developed as a modular Python package with GPU acceleration via PyTorch and RAPIDS ecosystem libraries. This approach ensures wide compatibility and usability across the remote sensing community.

Describe the main algorithms and how they have been implemented and parallelized:

TASK1. The core algorithm is a continuous change detection (CCD) approach that fits harmonic regression models to temporal segments of satellite imagery. The implementation follows a two-level parallelization strategy:
1. Coarse-grained parallelization: MPI is used to distribute batches of pixels across compute nodes, with each rank handling a subset of the total pixels.
2. Fine-grained parallelization: Within each node, threading is implemented using Python's multiprocessing to utilize all available cores for concurrent pixel processing.

The algorithm first partitions the input data (stored in HDF5 or NPY format) into equal-sized chunks. Each MPI rank loads its assigned chunks and further subdivides them among local threads. Results from all processes are aggregated and written to a single output PARQUET file using coordinated I/O operations.

TASK2.
1. Neural Network Approach: We will implement a U-Net architecture with temporal attention mechanisms, optimized for spatio-temporal segmentation. Each GPU processes independent tiles with results merged along overlap regions.

or

2. Graph-Based Approach: We will implement a CUDA-accelerated graph construction and community detection algorithm using the RAPIDS cuGraph library. The spatial data is partitioned into overlapping regions with pixels as nodes and similarities as edges. A GPU-accelerated maximum modularity algorithm identifies coherent patches. Multi-GPU parallelization is achieved by domain decomposition.

Both approaches leverage CUDA streams for overlapping computation and data transfer, maximizing GPU utilization.

Performance

Main performance bottlenecks:

TASK1. We have two main bottlenecks: I/O and compute. We need to read and process 500 million pixels. Since we cannot have the input pixels directly in memory, we need to read chunks of pixels from disk, which represents a bottleneck. In addition, each chunk processing involves fitting a Lasso regression, which represents a bottleneck in terms of compute. The LASSO regression in PyCCD is particularly demanding as it must be performed repeatedly as the algorithm identifies breakpoints in each time series.

TASK2.
The primary performance bottlenecks include:
1. Memory Constraints: Representing the full pixel graph for 500 million pixels would exceed available GPU memory, necessitating partitioning and potentially sub-optimal processing.
2. Data Loading: Transferring the 500GB parquet file to GPU memory creates a significant I/O bottleneck, particularly when multiple features per pixel are needed.
3. Graph Construction: For the graph-based approach, computing similarity metrics between pixels is computationally intensive.
4. Model Complexity: For the neural network approach, balancing model expressiveness with inference speed presents challenges for achieving real-time performance.
5. Result Merging: Combining results from partitioned data introduces overhead, especially when ensuring consistency along partition boundaries.

Describe possible solutions you have considered to improve the performance of the project:

TASK1. Upon profiling our code, we identified that the Lasso regression fitting was taking the most time among all processes. Therefore, we considered adapting the implementation of the Lasso regression available in the scikit-learn library to make it faster, which involves creating a custom Cython code for that purpose. Additionally, we plan to implement an optimized I/O strategy using parallel HDF5 and memory-mapped files to minimize disk bottlenecks. We're also exploring the use of numba-accelerated implementations of key numerical routines and developing a more efficient data chunking strategy that aligns with the locality patterns of the HPC system's storage hierarchy. For production the data has to be updated with new Sentinel-2 images but gains in efficiency are possible by re-using the outputs of the pixel-level time series analysis and just updating the most recent time segments.

TASK2. Implementing a multi-resolution approach that first identifies candidate regions at lower resolution before refining at full resolution. Developing specialized kernels for similarity computation that exploit the spatial locality of data. Implementing a pipeline that processes data in streams rather than loading the entire dataset at once. A strategy that could be very significant in terms of reducing the computational effort consists in discarding pixels for which the likelihood of change (estimated in TASK1) is very low. Graph approaches are particularly suitable to represent the remaining pixels (the ones with evidence of potential change) that typically do not exhibit a regular spatial pattern.

Describe the application enabling/optimization work that needs to be performed to achieve the target performance:

TASK1. To achieve our 24-hour processing target for 500 million pixels, we will implement:
1. I/O Optimization: Custom HDF5 chunking strategy and MPI-IO coordinated reads
2. If doable within the project span, specialized Cython implementation of LASSO regression leveraging BLAS libraries, replacing the general-purpose scikit-learn implementation.
3. Parallelization Enhancement
4. Checkpointing: Robust recovery mechanism to handle node failures without full restart.
5. Output Optimization for PARQUET
These optimizations will be implemented iteratively with benchmarking at each stage.

TASK2.
1. Implement GPU-optimized data loading pipelines using RAPIDS cuDF for efficient parquet processing.
2. Develop custom CUDA kernels for spatial similarity calculations, optimized for our specific feature vectors.
3. Optimize memory usage

Which computational performance limitations do you wish to solve with this project?:

TASK1. Our application consists in creating a national scale vegetation loss map, which means processing millions of pixels. In a normal computer this would be unfeasible, since the processing would take weeks to run. Therefore, with this project we expect to leverage the multiple nodes and CPUs to make our processing run much faster, in a reasonable amount of time. Specifically, we aim to overcome:
1. The sequential processing limitation that currently prevents real-time monitoring capabilities
2. The memory constraints that force excessive disk I/O operations when processing large geographic regions
3. If doable within the project span, the computational intensity of the LASSO regression fitting that dominates processing time

TASK2.
1. The scalability limitation of traditional remote sensing analysis pipelines that cannot handle country-scale data at high resolution.
2.The memory constraints of GPU-based graph algorithms when applied to massive geospatial datasets.
3. The computational intensity of spatio-temporal pattern detection across a very large set of time series data points.
4. The difficulty of maintaining spatial coherence when processing partitioned geographic data.
5. The balance between detection accuracy and processing speed needed for operational vegetation loss monitoring.

Application Support Team (AST)

Instructions: Not provided

Does your proposal require assistance from an AST on the selected partition(s)?: Yes

Please provide a short explanation why your proposal needs assistance from an AST:
At this point, needing assistance is mostly about combining MPI for I/O, namely for reading HDF5 files efficiently. We also may need assistance to find the optimal strategy for using all 128 processors for each node, managing available memory. For TASK2, we would benefit from GPU optimization support since we do not have experience with HPC GPU platforms.

Is your proposal a follow up of another submitted Epicure project to use EuroHPC quota for providing application support? : No

Select support types: Code GPU Support, Intra-node Optimization, Inter-node Optimization, GPU Optimization

Data Consent

Instructions: Not provided

space: Not provided

In case the proposal is awarded, EuroHPC JU would like to publish the Principal Investigator's and Team Members' names and organizations. This may involve sharing this information on our website, social media channels, or in other promotional materials related to the project. Please provide your consent below.: I consent

In order to submit the proposal, you must accept the terms and conditions stated in the Access Policy, hence confirming that you have read and understood the call procedures. The documentation can be found at https://eurohpc-ju.europa.eu/eurohpc-ju-call-proposals-development-access_en: I accept the terms and conditions stated in the Access Policy