

EGU24-7765, updated on 24 Apr 2024

<https://doi.org/10.5194/egusphere-egu24-7765>

EGU General Assembly 2024

© Author(s) 2024. This work is distributed under the Creative Commons Attribution 4.0 License.



## Unleashing the power of Dask with a high-throughput Trust Region Reflectance solver for raster datacubes

**Bernhard Raml**<sup>1</sup>, Raphael Quast<sup>1</sup>, Martin Schobben<sup>1</sup>, Christoph Reimer<sup>2</sup>, and Wolfgang Wagner<sup>1,2</sup>

<sup>1</sup>TU Wien, Geodesy and Geoinformation, Austria (bernhard.raml@geo.tuwien.ac.at)

<sup>2</sup>EODC Earth Observation Data Centre for Water Resources Monitoring GmbH

In remote sensing applications, the ability to efficiently fit models to vast amounts of observational data is vital for deriving high-quality data products, as well as accelerating research and development. Addressing this challenge, we developed a high-performance non-linear Trust Region Reflectance solver specialised for datacubes, by integrating Python's interoperability with C++ and Dask's distributed computing capabilities. Our solution achieves high throughput both locally and potentially on any Dask-compatible backend, such as EODC's Dask Gateway. The Dask framework takes care of chunking the datacube, and streaming each chunk efficiently to available workers where our specialised solver is applied. Introducing Dask for distributed computing enables our algorithm to run on different compatible backends. This approach not only broadens operational flexibility, but also allows us to focus on enhancing the algorithm's efficiency, free from concerns about concurrency. This enabled us to implement a highly efficient solver in C++, which is optimised to run on a single core, but still utilise all available resources effectively. For the heavy lifting, such as performing singular value decompositions and matrix operations we rely on Eigen, a powerful open-source C++ library specialized on linear algebra. To describe the spatial reference and other auxiliary data associated with our datacube, we employ the Xarray framework. Importantly, Xarray integrates seamlessly with Dask. Finally, to ensure robustness and extensibility of our framework, we applied state-of-the-art software engineering practices, including Continuous Integration and Test-Driven Development. In our work we demonstrate the significant performance gains achievable by effectively utilising available open-source frameworks, and adhering to best engineering practices. This is exemplified by our practical workflow demonstration to fit a soil moisture estimation model.