Middleware for Earth System Data

Julian Kunkel[1], Jakob Luettgau[1], Bryan N. Lawrence[2], Jens Jensen[2],
Sandro Fiore[4], Giuseppe Congiu[3], John Readey[5]
[1]Deutsches Klimarechenzentrum GmbH, [2]STFC Rutherford Appleton Laboratory,
[3]Seagate Technology LLC, [4]Euro-Mediterranean Center on Climate Change Foundation, [5]The HDF Group

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Abstract

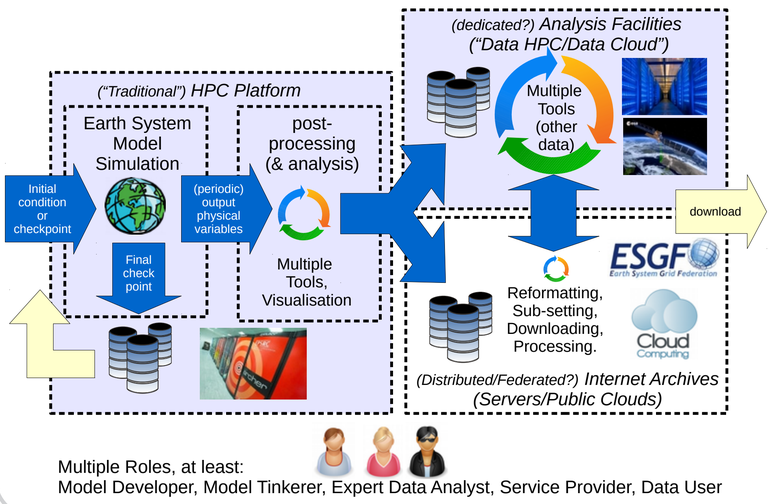
Making the best use of HPC in Earth simulation requires storing and manipulating vast quantities of data. Existing storage environments face usability and performance challenges for both domain scientists and the data centers supporting the scientists. These challenges arise from data discovery/access patterns, and the need to support complex legacy interfaces.

In the ESiWACE project, we develop a novel I/O middleware targeting, but not limited to, earth system data. Its architecture builds on well established end-user interfaces but utilizes scientific metadata to harness a data structure centric perspective.

Overview

- Center of excellence (ESiWACE)
  - Ongoing H2020 project
  - This work is part of WP4
- Goals of the earth system middleware
  - Accessing shared data with different APIs
    - NetCDF4 / HDF5
    - GRIB
  - Data-center optimized layout
    - Advanced data placement
    - Backends: object storage, file systems

Workflow



Multiple Roles, at least:
Model Developer, Model Tinkerer, Expert Data Analyst, Service Provider, Data User

Challenges

The middleware addresses a variety of challenges related to earth system applications:

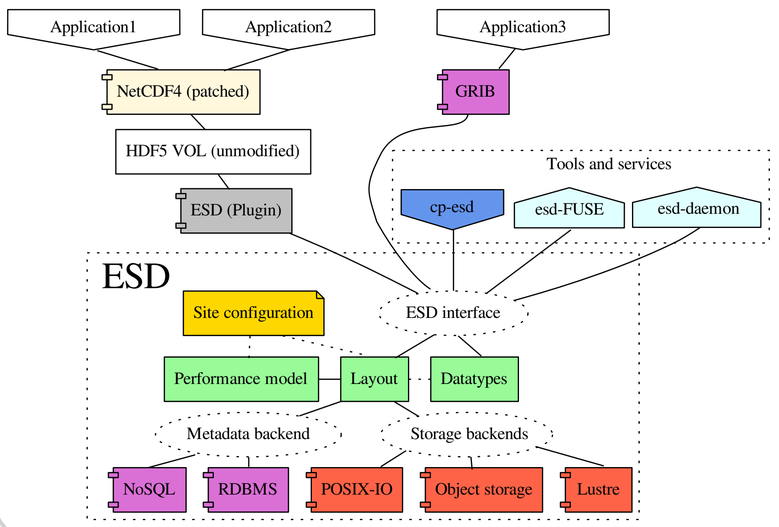
- Numerical climate and weather applications are very I/O intensive.
- Optimizing codes for specific supercomputers is usually not feasible.
- Libraries for standardized data description and optimized I/O (NetCDF, HDF5 or GRIB) do not adequately reflect system architectures.
- data management needs to scale with future data volumes in a way which provides acceptable data access latencies and data durability, and is cost-effective.
- support multi-disciplinary research through common APIs forflexible hybrid data and compute infrastructures.

Approach

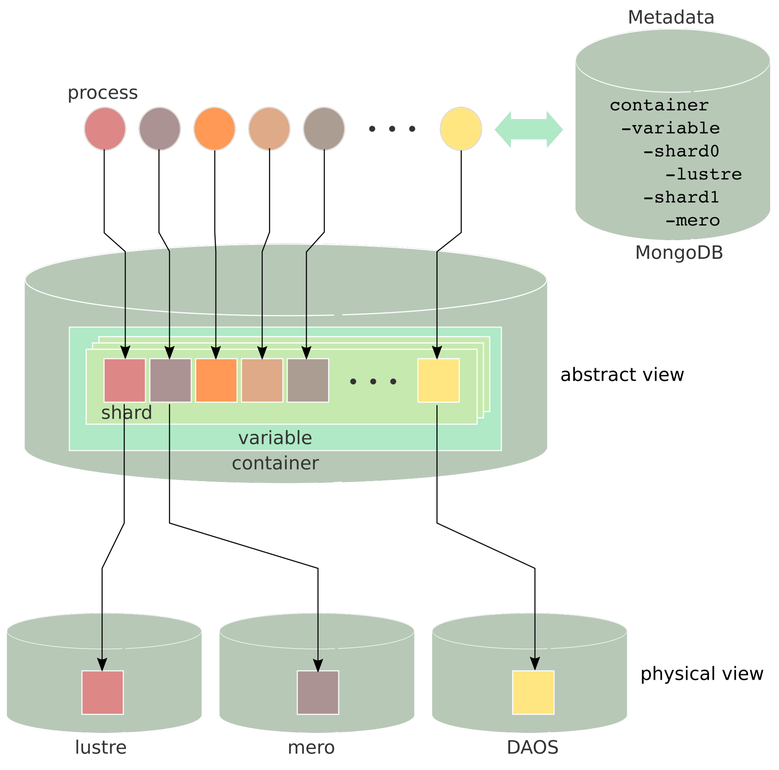
The *Earth System Data (ESD)* middleware is designed with the following objectives in mind:

1. Understand application data structure and scientific metadata, allow to expose using differen APIs (e.g., HDF5, NetCDF, GRIB).
2. Maps data structures to storage backends with different performance characteristics based on site specific configuration informed by a performance model.
3. Yields best write performance via op, and;timized data layout schemes that utilize elements from log-structured file systems.
4. Provides relaxed access semantics, tailored to scientific data generation for independent writes.
5. FUSE module for backwards compatibility and access via traditional APIs

Architecture Overview



Data Model



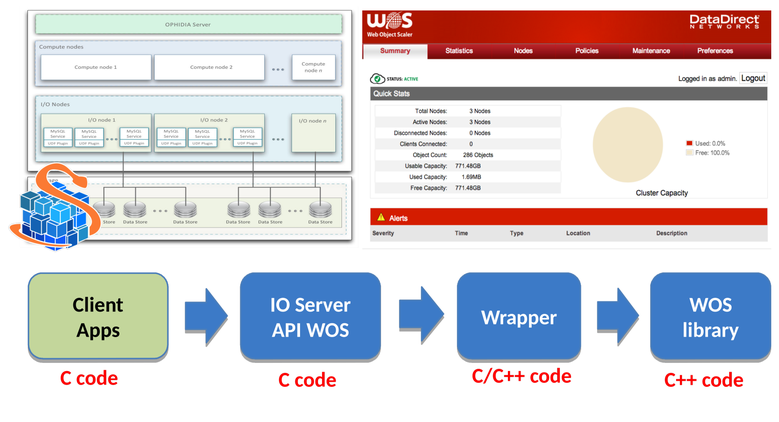
- Containers hold references to variables and metadata as well as different views/shards available for the same data.
- Variables describe the structure of the data including information about the dimensions.
- Metadata extends on semantical and technical aspects of data useful for ESDM and users.
- Shards hold the actual data, usually a shard is only a subdomain of the whole dataset.
- Sealed objects are immutable and protected by checksums, e.g. for completed research.
- Initially only 1) write-only and 2) read-only access is supported.

Semantics for Applications

Many of the ESDM changes are ment to be transparent to users. For advanced users new possiblities to interact with datasets are available. Usage sematics change as follows:

- Transparent for legacy MPI applications. No code changes required, but not always best performance.
- Different ways to access the data:
  - URL based access `ESD://` the the ability to query/subset/combine datasets.
  - FUSE filesystem, with configurable view.
  - Low-Level Library API for advanced users.
- Tools to simplify metadata and data organization and export. Ala `esdm-chmod`.
- Open close, write, epoch, job scheduler

Ophidia/In-memory analytics



- Layer for "fast" scientific data analytics.
- In-memory analytics for NWP/ESD
- Adapt activities regarding existing I/O analytics operators for NetCDF
- Implementation of two operators for GRIB
- Storage back-end for DDN WOS
  - 1st period: initial version of a C API and library for DDN WOS
  - Plan 2nd period: integration with the layout module & performance model

Summary

The ESD aids the interests of stakeholders: developers have less burden to provide system specific optimizations and can access their data in various ways. Data centers can utilize storage of different characteristics. We expect a working prototype with the core functionality within the next year. Following work will implement and fine-tune the cost model and layout component and provide additional backends.

Acknowledgments

The ESiWACE project received funding from the EU Horizon 2020 research and innovation programme under grant agreement No 675191. Disclaimer: This material reflects only the author's view and the EU-Commission is not responsible for any use that may be made of the information it contains.