Middleware for Earth System Data

Julian Kunkel\textsuperscript{1}  Jakob Luettgau\textsuperscript{1}  Bryan N. Lawrence\textsuperscript{2,3}  
Jens Jensen\textsuperscript{3}  Giuseppe Congiu\textsuperscript{4}  John Readey\textsuperscript{5}

\textsuperscript{1} German Climate Computing Center (DKRZ) 
\textsuperscript{2} NCAS, University of Reading 
\textsuperscript{3} STFC Rutherford Appleton Laboratory 
\textsuperscript{4} Seagate Technology LLC 
\textsuperscript{5} The HDF Group

PDSW-DISC, 2016-11-15
Outline

1 Introduction

2 Approach

3 Roadmap

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
Challenges in the domain of climate/weather

- High data volume and velocity
- Data management practice does not scale
  - e.g., hierarchical namespaces do not reflect use cases
  - Scientists spend quite some time to define the namespace
- Suboptimal performance (& perf. portability) of data formats
  - Tuning for NetCDF, HDF5, and GRIB necessary
  - Scientists worry about interoperability
- Data conversion is often needed
  - Especially between NetCDF and GRIB
  - To combine data from multiple experiments, time steps, ...
- External data services to share data in the community
  - (Scientific) metadata is provided by databases
Design Goals of the Earth System Data Middleware

1. Understand application data structures and scientific metadata
2. Flexible mapping of data to multiple storage backends
3. Placement based on site-configuration + performance model
4. Site-specific optimized data layout schemes
5. Relaxed access semantics, tailored to scientific data generation
6. A configurable namespace based on scientific metadata
Architecture

Application1

NetCDF4 (patched)

HDF5 VOL (unmodified)

ESD (Plugin)

Application2

Application3

GRIB

Tools and services

cp-esd
esd-FUSE
esd-daemon

ESD

Site configuration

Performance model

Layout

Datatypes

Metadata backend

Storage backends

NoSQL
RDBMS
POSIX-IO
Object storage
Lustre
Benefits

- Expose/access the same data via different APIs
- Independent and lock-free writes from parallel applications
- No fixed storage layout
- Less performance tuning from users needed
- Exploit characteristics of different storage technology
- Multiple layouts of one data structure optimize access patterns
- Flexible namespace (similar to MP3 library)

---

1 To achieve portability, we provide commands to create platform-independent file formats on the site’s boundary/long-term archive.
Roadmap

- Done: Example HDF5 VOL (for understanding)
- 75%: HDF5 plugin for Seagate Object Store technology
- Done: High-level design
- 75%: Log-structured file mapping for POSIX backend
- Next: Datatypes, one storage backend, manual layout
- Q4 2017: Prototype for the system architecture
- Q4 2018: Production version with mappings for different sites
The ESiWACE project has received funding from the European Union’s 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