Exploiting Weather & Climate Data at Scale (WP4)

Julian Kunkel[1] Bryan N. Lawrence[2,3] Jakob Luettgau[1] Neil Massey[4] Alessandro Danca[5] Sandro Fiore[5] Huang Hu[6]

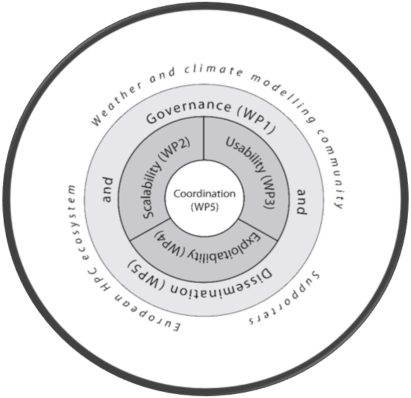
1 German Climate Computing Center (DKRZ)
2 UK National Centre for Atmospheric Science
3 Department of Meteorology, University of Reading
4 STFC Rutherford Appleton Laboratory
5 CMCC Foundation
6 Seagate Technology LLC

ESiWACE GA, Dec 2017

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

Outline

Project Organisation



WP1 Governance and Engagement

WP2 Global high-resolution model demonstrators

WP3 Usability

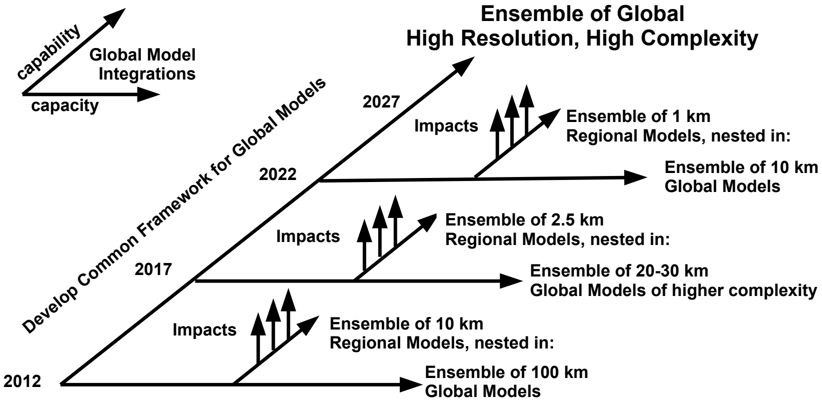
WP4 Exploitability

- The business of storing and exploiting high volume data
- New storage layout for Earth system data
- New methods of exploiting tape

WP5 Management and Disssemination

Community Goals

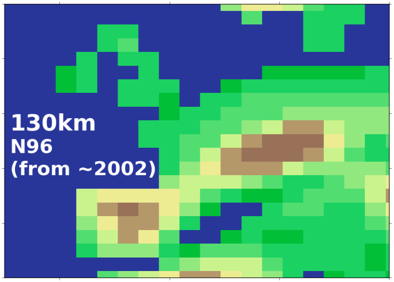
This is what we said in 2012:

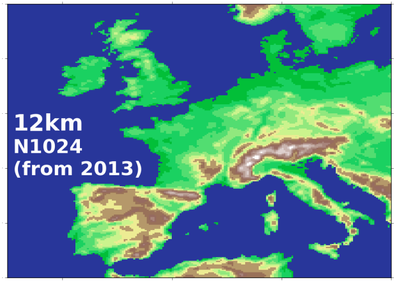


. . . consistent with needing an exascale machine!

A modest (?) step ...

Europe within a global model ...





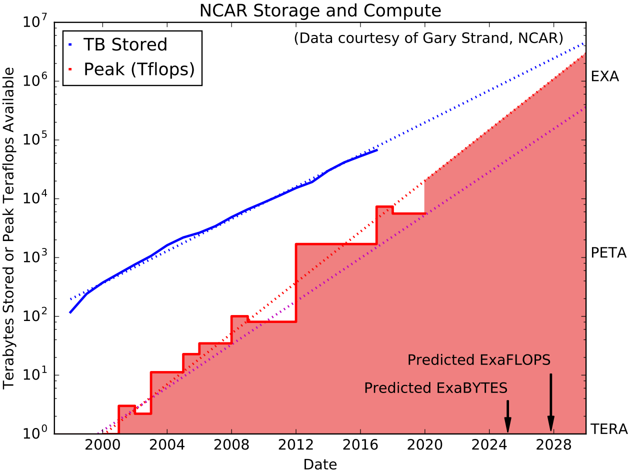
One "field-year" — 26 GB

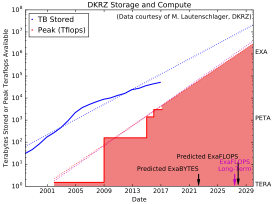
1 field, 1 year, 6 hourly, 80 levels
1 x 1440 x 80 x 148 x 192

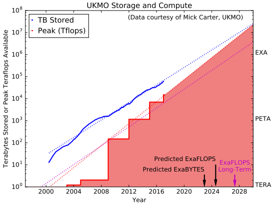
One "field-year" — >6 TB

1 field, 1 year, 6 hourly, 180 levels
1 x 1440 x 180 x 1536 x 2048

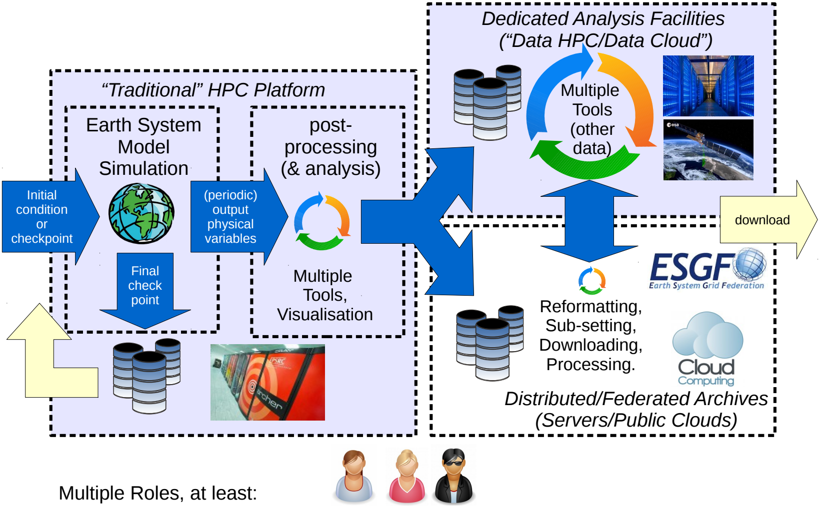
. . . towards Exascale







Heterogeneity in the Workflow Environment



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

Issues and Actions

Issues

- Cost: Disk prices not falling as fast as they used to.
- Behaviour: Larger groups sharing data for longer ⇒ data is re-used for longer.
- Performance: Traditional POSIX file systems not scalable for shared access.
- Software: Little software for our domain which can exploit object storage and use the public cloud.
- Tape: Tape remains important, particularly for large amounts of **cold** data.

Issues and Actions

Issues

- Cost: Disk prices not falling as fast as they used to.
- Behaviour: Larger groups sharing data for longer ⇒ data is re-used for longer.
- Performance: Traditional POSIX file systems not scalable for shared access.
- Software: Little software for our domain which can exploit object storage and use the public cloud.
- Tape: Tape remains important, particularly for large amounts of **cold** data.

ESiWACE Actions

- Better understanding of costs and performance of existing and near-term storage technologies.
- Earth System Middleware prototype — provides an interface between the commonly used NetCDF/HDF library and storage which addresses the performance of POSIX and the usability of object stores (and more).
- Semantic Storage Library prototype: — Python library that uses a "weather/climate" abstraction (CF-NetCDF data model) to allow one "file" to be stored across tiers of, e.g. POSIX disk, object store, and tape.

Work Package 4 — Exploitability (of data)

Partners

DKRZ, STFC, CMCC, Seagate, UREAD

ECMWF was originally a partner, but we removed the relevant task in the reprofiling following the review

Task 4.1

Cost and Performance

Documentation
Formal deliverable produced, ongoing work for publication and dissemination.

Task 4.2

New Storage Layout

Software
ESD Middleware
Formal software design delivered, work on backends underway.

Task 4.3

New Tape Methods

Software
Semantic Storage Lib
Prototype pieces in place.

Methodology

Simple models

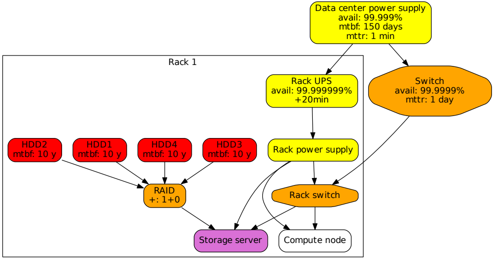
High-level representation of hardware / software components.

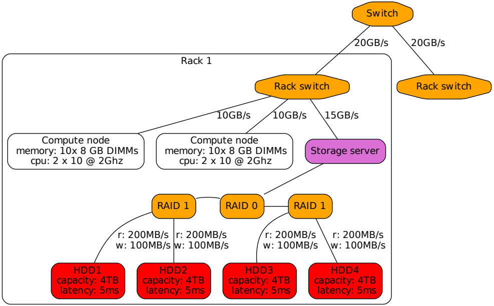
Includes:

- performance,
- reslience
- cost

Deliverable

Scenarios discussing architectural changes for data centres, and implications for cost/performance





Performance Modelling

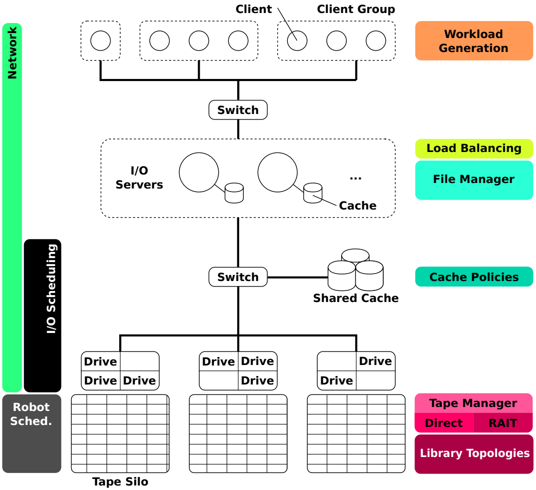
Detailed Modelling

A simulator has been developed, covering:

- Hardware, software: tape drives, library, cache
- Can replay recorded FTP traces
- Validated with DKRZ environment

Usage

Aim to use to evaluate performance and costs of future storage scenarios.

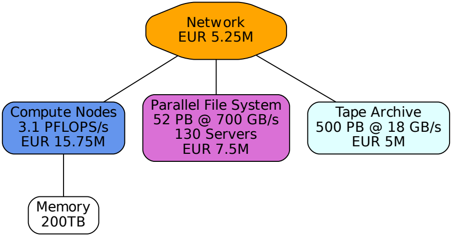


Some Results

Costs of storage for DKRZ

- Tape: 12 € per TB/ year
- Software licenses for tape are driving the costs!
- Parallel Disk: 28(36) € TB/year
- Object storage: 12.5 € TB/year (without software license costs)
- Cloud: 48 $ TB/year (only storage, access adds costs)
- Idle (unused) data is an important cost driver!

Can consider various scenarios

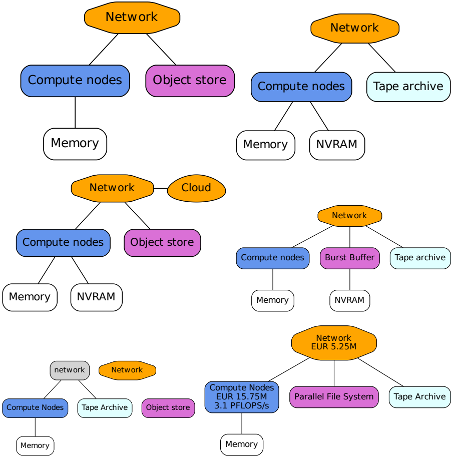


Some Results

Costs of storage for DKRZ

- Tape: 12 € per TB/ year
- Software licenses for tape are driving the costs!
- Parallel Disk: 28(36) € TB/year
- Object storage: 12.5 € TB/year (without software license costs)
- Cloud: 48 $ TB/year (only storage, access adds costs)
- Idle (unused) data is an important cost driver!

Can consider various scenarios



We have yet to work through all of these (and others)!

The problem space in more detail

Challenges in the domain of climate/weather

- Large data volume and high velocity
- Data management practice does not scale & not portable
  - Cannot easily manage file placement and knowledge of what file contains.
  - Hierarchical namespaces does not reflect use cases.
  - Bespoke solutions at every site!
- Suboptimal performance & performance portability
  - Cannot properly exploit the hardware / storage landscape
  - Tuning for file formats and file systems necessary at the *application* level
- Data conversion is often needed
  - To combine data from multiple experiments, time steps, ...

The problem space in more detail

Challenges in the domain of climate/weather

- Large data volume and high velocity
- Data management practice does not scale & not portable
  - Cannot easily manage file placement and knowledge of what file contains.
  - Hierarchical namespaces does not reflect use cases.
  - Bespoke solutions at every site!
- Suboptimal performance & performance portability
  - Cannot properly exploit the hardware
  - Tuning for file formats and file systems necessary at the *application* level
- Data conversion is often needed
  - To combine data from multiple experiments, time steps, ...

Approach

Design Goals of the Earth System Data Middleware

1. Reduce penalties of **shared** file access
2. Ease of use and deployment
3. Understand application data structures and scientific metadata
4. Flexible mapping of data to multiple storage backends
5. Placement based on site-configuration and *limited* performance model
6. Site-specific (optimized) data layout schemes
7. Relaxed access semantics, tailored to scientific data generation
8. A configurable namespace based on scientific metadata

Approach (continued)

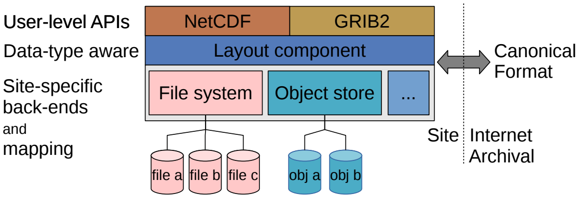
Expected Benefits

- Expose/access the same data via different APIs
- Independent and lock-free writes from parallel applications
- Storage layout is optimised to local storage
  - Exploits characteristics of storage, rather than one size stream of bytes fits all.
  - To achieve portability, we provide commands to create platform-independent file formats on the site boundary or for use in the long-term archive (see also the SemSL).
- Less performance tuning from users needed
- One data structure can be fully or partially replicated with different layouts to optimize access patterns
- Flexible namespace (similar to MP3 library)

Architecture

Key Concepts

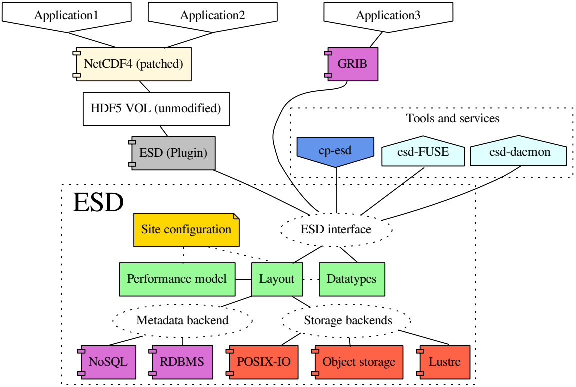
- Applications work through existing (NetCDF library) Other interfaces could be supported in the future.
- New middleware between HDF library andstorage exposes information to a "layout component" about the available storage, and data is fragmented accordingly.
- Data is then written efficiently.



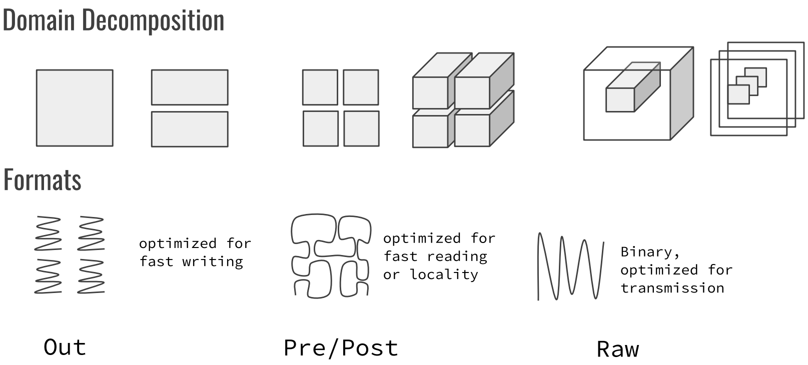
Architecture

Key Concepts

- Applications work through existing (NetCDF library) Other interfaces could be supported in the future.
- New middleware between HDF library andstorage exposes information to a "layout component" about the available storage, and data is fragmented accordingly.
- Data is then written efficiently.

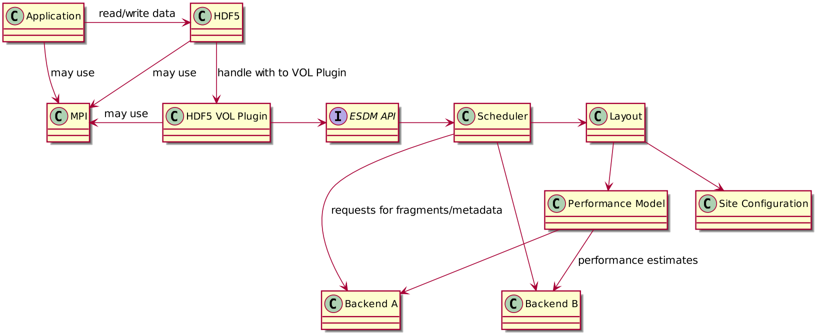


Optimizing for Data Representation



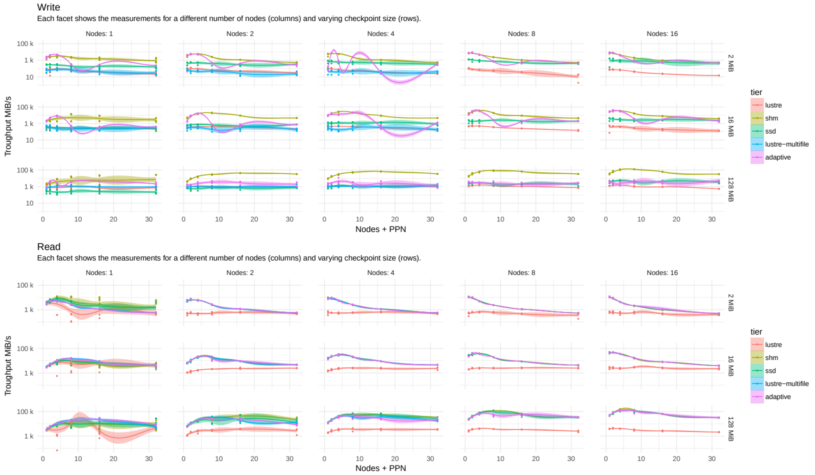
Storage makes placement decisions exploiting the storage landscape

Backend Specific Optimization



Interplay of a IO scheduler, a layout component and storage specific performance models.

First Results with POSIX backend

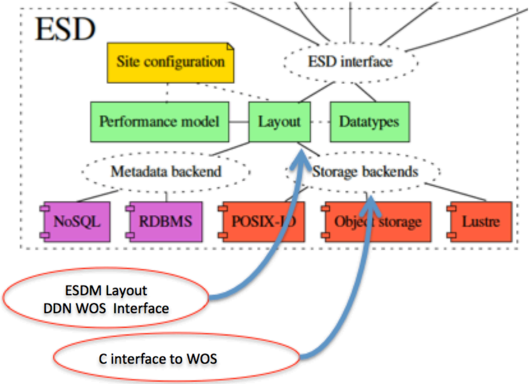


Adaptive Tier Selection for HDF5/NetCDF without requiring changes to existing applications. (SC17 Research Poster).

Other backends – DDN Object Store (CMCC)

WOS Progress

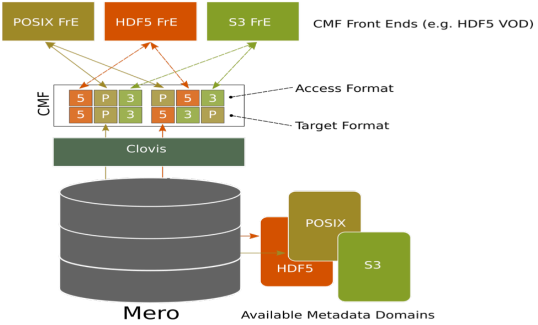
- First draft of layout interfaces.
- Developed C wrapper for the C++ DDN WOS libraries with a direct mapping.
- Designed a parallel approach for independent/multiple write operations on WOS storage.



Other backends – Seagate CLOVIS

Seagate Progress

- Data structures and interfaces designed for ESDM to access objects in Mero with Clovis APIs.
- First draft code.
- Read & write block-aligned regions from Mero cluster via ESDM requests, in parallel.
- In the future: Seagate will be working on optimisation, performance and scalability in stable code.

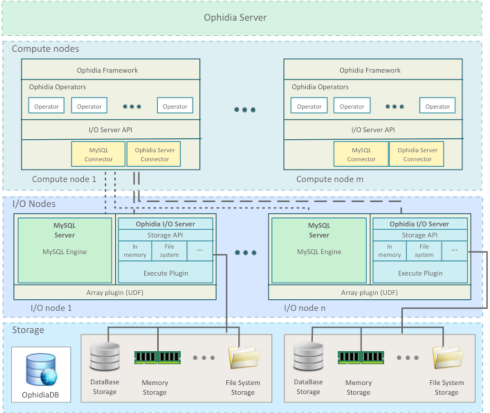


Deployment Testing Example

Test and Deployment

Ophidia as a test application for ESDM

- **Import** and **Export** Ophidia operators adapted for integration with ESDM storage
- **In-memory** data analysis benchmark using ESDM



GRIB support

- Extend Ophidia import/ export operators to provide GRIB support (implementation expected next year).

ESDM Status & Roadmap

- Done: ESDM Architecture Design for Prototype
- Done: Proof of concept for adaptive tier selection
- 70%: HDF5 VOL Plugin as Application to ESDM Adapter
- 30%: ESDM Core Implementation as Library
- 20%: Backend Plugins for POSIX, Clovis, WOS
- Q1 2018: Backend for POSIX, Metadata in MongoDB
- Q1 2018: Benchmarking at sites
- Q2 2018: Backends for Clovis, WOS
- Q4 2018: Production version with site-specific mappings

The problem space in more detail

Challenges in the domain of climate/weather

- Large data volume and high velocity
- Data management practice does not scale & not portable
  - Cannot easily manage file placement and knowledge of what file contains.
  - Hierarchical namespaces does not reflect use cases.
  - Bespoke solutions at every site!
- Suboptimal performance & performance portability
  - Cannot properly exploit the hardware
  - Tuning for file formats and file systems necessary at the *application* level
- Data conversion is often needed
  - To combine data from multiple experiments, time steps, ...

Approach

Design Goals of the Semantic Storage Library

1. Provide a portable library to address user management of data files on disk and tape which
   - does not *require* significant sysadmin interaction, but
   - can make use of local customisation if available/possible.
2. ~~Increase bandwidth to/from tape by exploiting RAID-to-TAPE~~.
3. Exploit current and likely storage architectures (tape, disk caches, POSIX and object stores).
4. Can be deployed in prototype fast enough that we can use it for the Exascale Demonstrator.
5. Exploit existing metadata conventions.
6. Can eventually be backported to work with the ESDM.

Architecture

CFA Framework (`https://goo.gl/DdxGtw`)

1. Based on CF Aggregation framework proposed 6 years ago `https://goo.gl/K8jCP8`.
2. Define how multiple CF fields may be combined into one larger field (or how one large field can be divided).
3. Fully general and based purely on CF metadata.
4. Includes a syntax for storing an aggregation in a NetCDF file using **JSON** string content to point at aggregated files.

Architecture

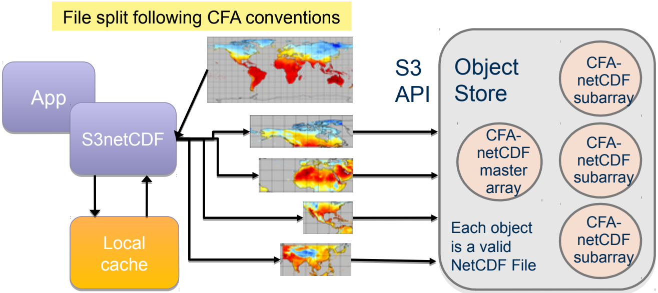
CFA Framework (`https://goo.gl/DdxGtw`)

1. Based on CF Aggregation framework proposed 6 years ago `https://goo.gl/K8jCP8`.
2. Define how multiple CF fields may be combined into one larger field (or how one large field can be divided).
3. Fully general and based purely on CF metadata.
4. Includes a syntax for storing an aggregation in a NetCDF file using **JSON** string content to point at aggregated files.

Two Key Components

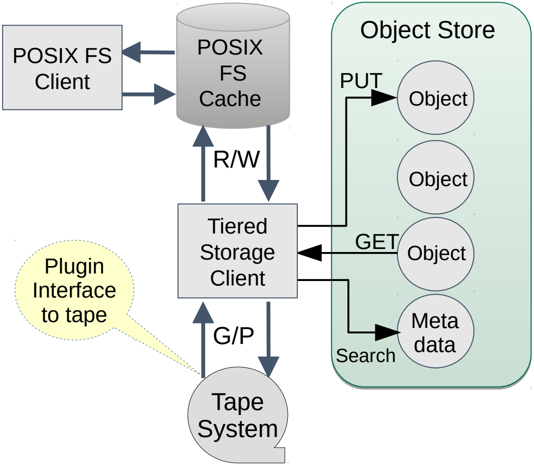
1. S3NetCDF — a drop in replacement for NetCDF4-python.
2. CacheFace - a portable drop-in cache with support for object stores and tape systems.

S3NetCDF (working title)



- Master Array File is a NetCDF file containing dimensions and metadata for the variables (including URLs to fragment file locations).
- Master Array File can be in persistent memory or online, nearline, etc
- NetCDF tools can query file CF metadata content without fetching them.
- *Currently serial, work on parallelisation underway.*

CacheFace (working title)



CacheFace Status

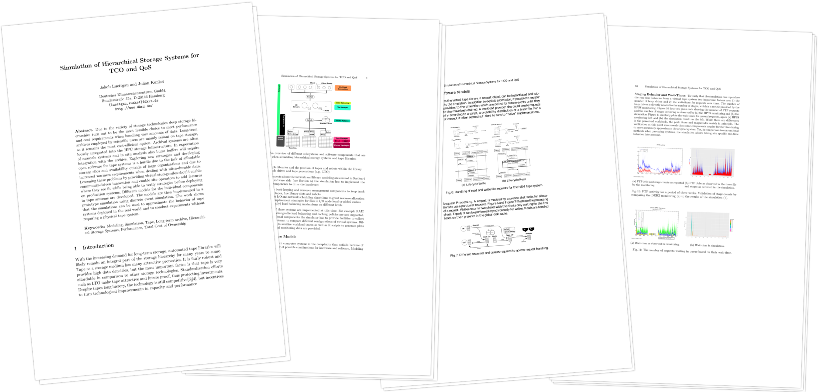
Prototype pieces exit

- Simple metadata system designed.
- Cache system designed and prototype built that can use Minio interface to object store.
- Another cache system built which depends on our bespoke tape environment (ElasticTape).
- Work planned on integration and developing plugin concept.

Dissemination and Publications

- SC16 Research Poster
  - Modeling and Simulation of Tape Libraries for Hierarchical Storage Systems
- PDSW-DISCS Workshop at SC16 WiP
  - Middleware for Earth System Data
- HPC-IODC Workshop at ISC17 Paper
  - Simulation of Hierarchical Storage Systems for TCO and QoS
- ISC17 Project Poster
  - Middleware for Earth System Data
- PDSW-DISCS Workshop at SC17 WiP
  - Towards Structure-Aware Earth System Data Management
- SC17 Research Poster
  - Adaptive Tier Selection for NetCDF/HDF5

HPC-IODC Workshop at ISC17



Simulation of tape archives to improve hierarchical storage systems and test novel integration of cold storage in data centers.

PDSW-DISCS Workshop at SC17

Towards Structure-Aware Earth System Data Management

Jakob Luettgau
German Climate Computing Center
luettgau@dkrz.de

Julian Kunkel
German Climate Compute Center
kunkel@dkrz.de

Bryan N. Lawrence
University of Reading
bryan.lawrence@ncas.ac.uk

Sandro Fiore
CMCC Foundation
sandro.fiore@cmcc.it

Huang Hua
Seagate Technology LLC
hua.huang@seagate.com

ABSTRACT

Current storage environments confront domain scientist and data center operators with usability and performance challenges. To achieve performance portability data description libraries such as HDF5 and NetCDF are widely adopted. At the moment, these libraries struggle to adequately account for access patterns when reading and writing data to multi-tier distributed storage systems. As part of the ESiWACE[1] project, we develop a novel I/O middleware targeting, but not limited to, earth system data. The architecture builds on top of well-established end-user interfaces but utilizes scientific metadata to harness a data structure centric perspective.

1 INTRODUCTION

As scientists are adapting their codes to take advantage of the next-generation exascale systems, the I/O bottleneck becomes a major challenge[1–3] because storage systems struggle to absorb data at the same pace as it is generated. Especially, simulation codes such as climate and numerical weather prediction periodically experience bursty I/O, as they are writing so called checkpoints to achieve fault tolerance and for data analysis. Technological and budgetary constraints have led to complex storage hierarchies.

2 APPROACH

The overall architecture of the *Earth System Data (ESD)* middleware is depicted in Figure 1. It is designed to address multiple I/O challenges, in particular this includes:

(1) awareness of application data structures and scientific metadata, which lets us expose the same data via different APIs;
(2) map data structures to storage backends based on performance characteristics of storage configuration of site
(3) optimize for write performance by combining data fragmentation and elements from log-structured file systems;
(4) provides relaxed access semantics, tailored to scientific data generation for independent writes, and;

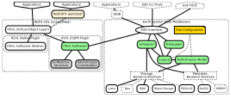


Figure 1: Overview of the architecture, which allows the middleware optimize for site specific data services without requiring changes to applications.

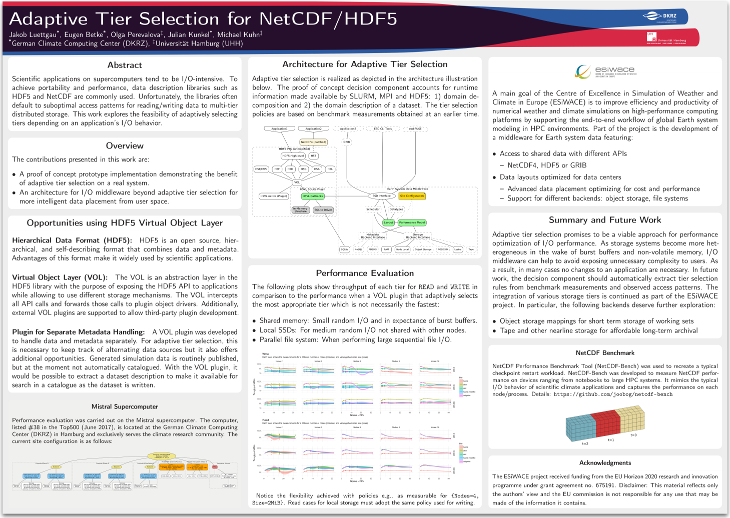
3 ACTIVITIES AND STATUS

A first prototype was developed demonstrating the viability of adaptively choosing tiers based on policies to achieve performance gains. By using information exposed by HDF5, MPI and SLURM it was possible to account for checkpoint size and domain decomposition (nodes+ppn). Without changing application code interfacing with NetCDF/HDF5, it was possible to choose different I/O paths (e.g. SHM, SSD, Lustre) for different requests. In addition, the design of the proposed architecture of the middleware thoroughly documented covering access semantics, (meta)data backends and processing throughout the I/O stack. The full report is available on the ESiWACE website [1]. Currently, the preliminary interfaces for the data and metadata backends are being developed.

4 SUMMARY

The ESDM addresses the challenges of multiple 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 coming year. Following work will implement and fine-tune the cost model and layout component and provide additional backends. Besides storage backends, an integration of scientific workflows with workload manager requires

SC17 Research Poster



Adaptive Tier Selection for NetCDF/HDF5

Jakob Luettgau*, Eugen Betke*, Olga Perevalova‡, Julian Kunkel*, Michael Kuhn‡
*German Climate Computing Center (DKRZ), ‡Universität Hamburg (UHH)

Abstract

Scientific applications on supercomputers tend to be I/O-intensive. To achieve portability and performance, data description libraries such as HDF5 and NetCDF are commonly used. Unfortunately, the libraries often default to suboptimal access patterns for reading/writing data to multi-tier distributed storage. This work explores the feasibility of adaptively selecting tiers depending on an application's I/O behavior.

Overview

The contributions presented in this work are:

- A proof of concept prototype implementation demonstrating the benefit of adaptive tier selection on a real system.
- An architecture for I/O middleware beyond adaptive tier selection for more intelligent data placement from user space.

Opportunities using HDF5 Virtual Object Layer

Hierarchical Data Format (HDF5): HDF5 is an open source, hierarchical, and self-describing format that combines data and metadata. Advantages of this format make it widely used by scientific applications.

Virtual Object Layer (VOL): The VOL is an abstraction layer in the HDF5 library with the purpose of exposing the HDF5 API to applications while allowing to use different storage mechanisms. The VOL intercepts all API calls and forwards those calls to plugin object drivers. Additionally, external VOL plugins are supported to allow third-party plugin development.

Plugin for Separate Metadata Handling: A VOL plugin was developed to handle data and metadata separately. For adaptive tier selection, this is necessary to keep track of alternating data sources but it also offers additional opportunities. Generated simulation data is routinely published, but at the moment not automatically catalogued. With the VOL plugin, it would be possible to extract a dataset description to make it available for search in a catalogue as the dataset is written.

Mistral Supercomputer

Performance evaluation was carried out on the Mistral supercomputer. The computer, listed #38 in the Top500 (June 2017), is located at the German Climate Computing Center (DKRZ) in Hamburg and exclusively serves the climate research community. The current site configuration is as follows:

Architecture for Adaptive Tier Selection

Adaptive tier selection is realized as depicted in the architecture illustration below. The proof of concept decision component accounts for runtime information made available by SLURM, MPI and HDF5: 1) domain decomposition and 2) the domain description of a dataset. The tier selection policies are based on benchmark measurements obtained at an earlier time.

Performance Evaluation

The following plots show throughput of each tier for READ and WRITE in comparison to the performance when a VOL plugin that adaptively selects the most appropriate tier which is not necessarily the fastest:

- Shared memory: Small random I/O and in expectance of burst buffers.
- Local SSDs: For medium random I/O not shared with other nodes.
- Parallel file system: When performing large sequential file I/O.

Notice the flexibility achieved with policies e.g., as measurable for {Nodes=4, Size=2MiB}. Read cases for local storage must adopt the same policy used for writing.

A main goal of the Centre of Excellence in Simulation of Weather and Climate in Europe (ESiWACE) is to improve efficiency and productivity of numerical weather and climate simulations on high-performance computing platforms by supporting the end-to-end workflow of global Earth system modeling in HPC environments. Part of the project is the development of a middleware for Earth system data featuring:

- Access to shared data with different APIs
  - NetCDF4, HDF5 or GRIB
- Data layouts optimized for data centers
  - Advanced data placement optimizing for cost and performance
  - Support for different backends: object storage, file systems

Summary and Future Work

Adaptive tier selection promises to be a viable approach for performance optimization of I/O performance. As storage systems become more heterogeneous in the wake of burst buffers and non-volatile memory, I/O middleware can help to avoid exposing unnecessary complexity to users. As a result, in many cases no changes to an application are necessary. In future work, the decision component should automatically extract tier selection rules from benchmark measurements and observed access patterns. The integration of various storage tiers is continued as part of the ESiWACE project. In particular, the following backends deserve further exploration:

- Object storage mappings for short term storage of working sets
- Tape and other nearline storage for affordable long-term archival

NetCDF Benchmark

NetCDF Performance Benchmark Tool (NetCDF-Bench) was used to recreate a typical checkpoint restart workload. NetCDF-Bench was developed to measure NetCDF performance on devices ranging from notebooks to large HPC systems. It mimics the typical I/O behavior of scientific climate applications and captures the performance on each node/process. Details: https://github.com/joobog/netcdf-bench

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 authors' view and the EU commission is not responsible for any use that may be made of the information it contains.

Proof of concept and early work on site characterization.

The landscape is (rapidly) changing

Environment

Scheduler support for Non Volatile Memory. Different modes of use. (NEXTGENIO)

Dynamic on the fly file systems (BeeOND, ADA FS, even CephFS) . . .

NetCDF

Ongoing proposals to address thread safety etc - NetCDF will evolve.

Considering NCX format for optimising READ-only access. Not HDF (but alongside HDF).

HDF

H5Serv and HSDS/HDF-Cloud

Serving files via REST (storage can be files or fragments), API unchanged.
(We are experimenting with this in ESIWACE1 WP4)

ExaHDF

Multiple formats under HDF5 AP (ADIOS, NC3 etc)

Climate use case. Better HPC performance, asyncio, data model support for cloud resolving model grids . . .

Supporting the EU Exascale Vision and Beyond

To demonstrate benefits, we need better integration with WPs

Short Term Goals

- "ESiWACE1 Hero Runs" should send (some) data to JASMIN for complete workflow proof of principle.
- Data should then be fragmented using the SemSL so that some data is on tape and some on disk, and **users** can control what is where.

Medium Term Goals

- Utilise the ESDM inside a large model run (WP2)
- Consider how to connect ESDM output with WAN transfer and SemSL in workflow. (WP3)
- Consider ESDM integration with other on the fly file systems, ExaHDF etc (we can do that)
- How will we work with the ESD sites?

- We will need to work out how to establish appropriate internal liaisons to make most of those things happen.
- We would like to have some active discussion about how to take this forward !

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