Department of Computer Science

University of Reading

Fighting the Data Deluge with Data-Centric Middleware

Limitless Storage

Limitless Possibilities

https://aces.cs.reading.ac.uk

https://hps.vi4io.org

Julian M. Kunkel, Bryan Lawrence

PASC Minisymposium: The Exabyte Data Challenge

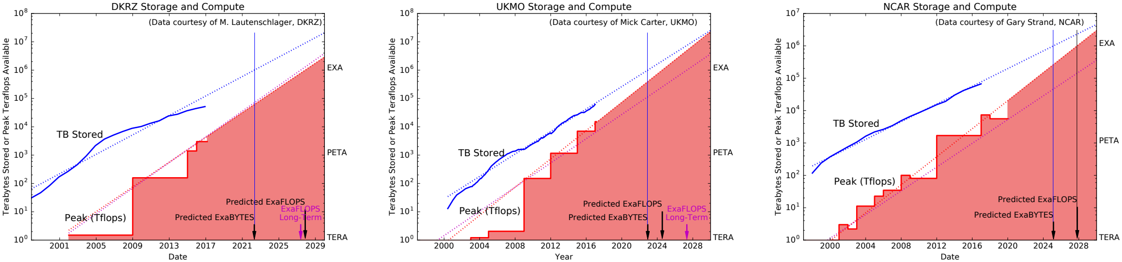
2019-06-14

Copyright University of Reading

LIMITLESS **POTENTIAL** | LIMITLESS **OPPORTUNITIES** | LIMITLESS **IMPACT**

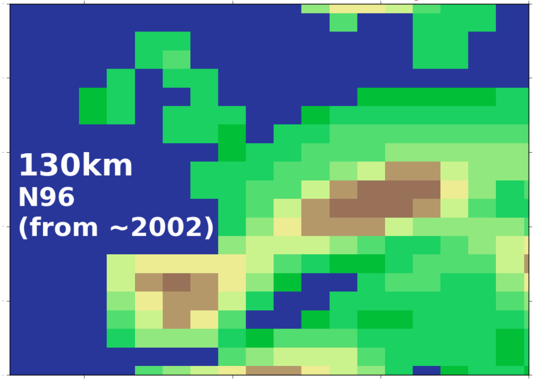
Outline

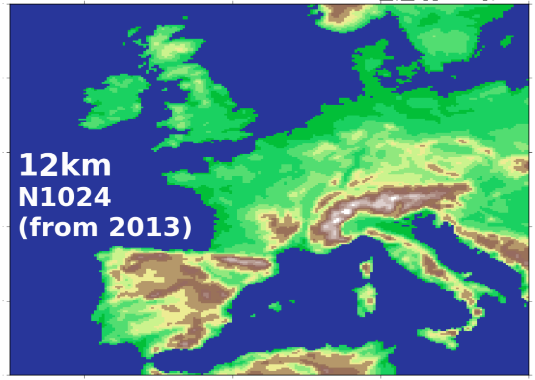
The Exabyte Challenge in Climate and Weather



Long-term predictions uses historical data (before 2000)

Volume: A Modest (?) Step ...





One "field-year": 26 GB

1 field, 1 year, 6 hourly, 80 levels

1 x 1440 x 80 x 148 x 192

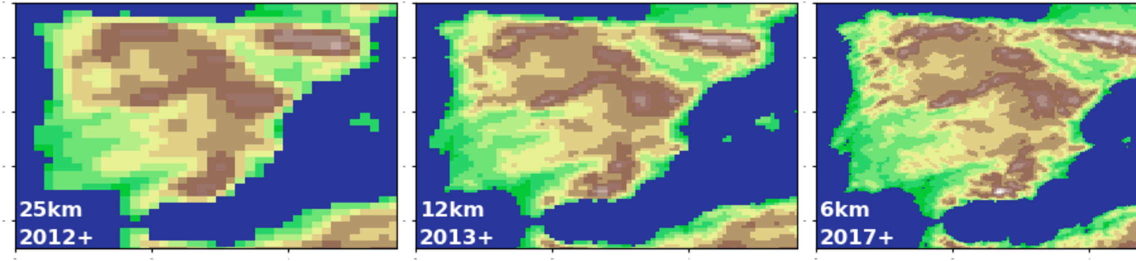
One "field-year": 6 TB

1 field, 1 year, 6 hourly, 180 levels

1 x 1440 x 180 x 1536 x 2048

Volume — The Reality of Global 1km Grids

University of Reading



1 km is the current European Network for Earth System Modelling (ENES) goal!

Consider N13256 (1.01km, 26512x19884):

- 1 field, 1 year, 6 hourly, 180 levels
- 1 x 1440 x 180 x 26512 x 19884 = 1.09 PB
- but with 10 variables hourly: > 220 TB/day!

Can no longer consider serial diagnostics

Climate/Weather Workflows

General Challenges Related to IO

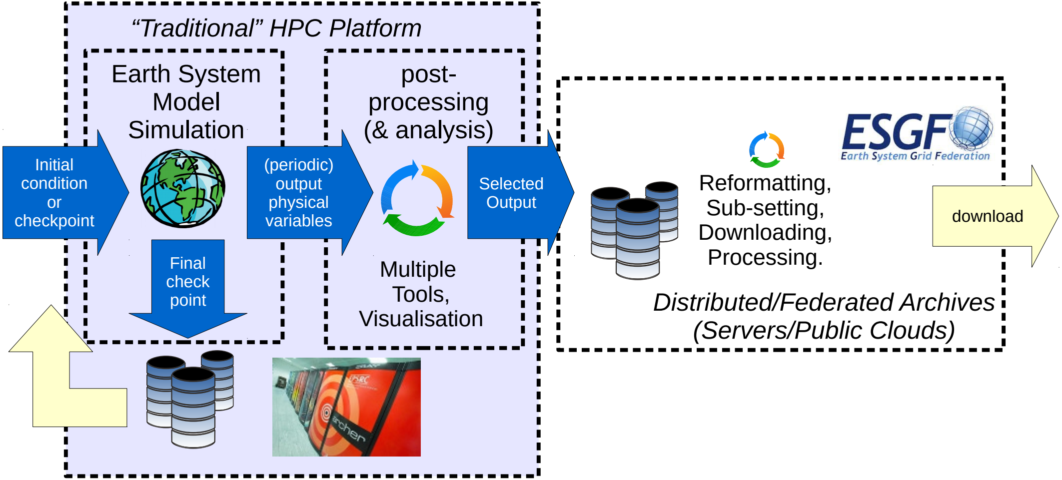
- Programming of efficient workflows
- Efficient analysis of data
- Organizing data sets
- Ensuring reproducability of workflows/provenance of data
- Meeting the compute/storage needs in future complex hardware landscape

Expected Data Characteristics in 2020+

- Velocity: Input 5 TB/day (for NWP; reduced data from instruments)
- Volume: Data output of ensembles in PBs of data
- Variety: Various file formats, input sources
- Usability: Data products are widely used by 3rd parties

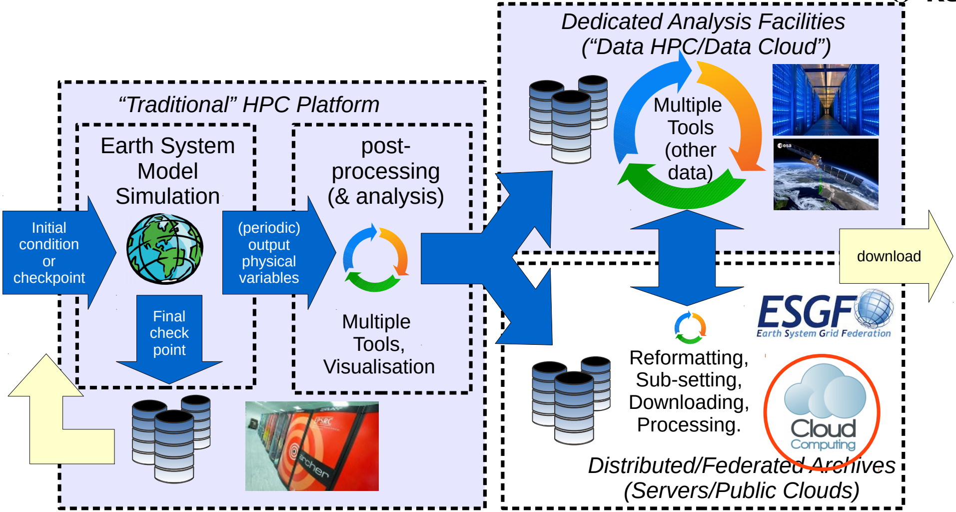
How we used to do it: From Supercomputer to Download

University of Reading



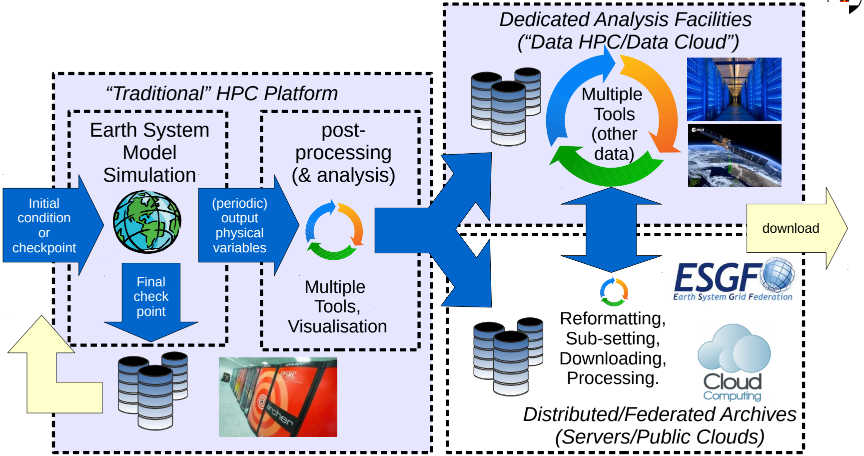
Many different supercomputing environments

University of Reading



Many different supercomputing environments

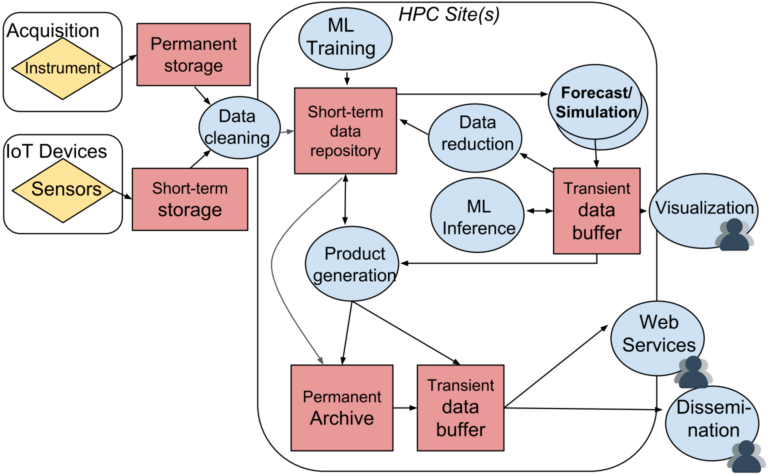
University of Reading



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

Smarter Climate/Weather Workflows in 2020+

University of Reading



- IoT (and mobile devices)
  - Additional data provider
  - Improves short-term weather prediction
- Machine learning support
  - Localize known patterns
  - Interactive use
    Visual analytics
- Data reduction
  - Output is triggered by events (ML)
  - Compress data of ensembles

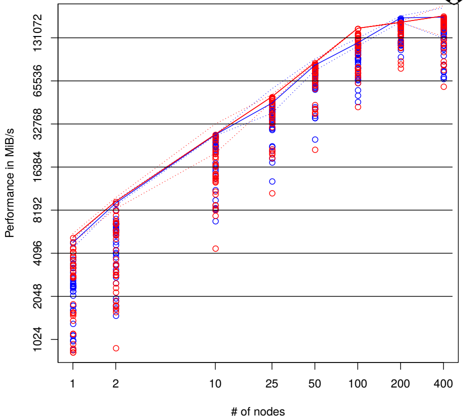
General I/O Challenges

- Large data volume and high velocity
- Data management practice does not scale and is not portable
  - Cannot easily manage file placement and knowledge of what file contains
  - Hierarchical namespaces does not reflect use cases
  - Individual strategies at every site
- Data conversion/merging is often needed
  - To combine data from multiple experiments, time steps, ...
- The storage stack becomes more inhomogeneous
  - Non-volatile memory, SSDs, HDDs, tape
  - Node-local, vs. global shared, partial access (e.g., racks)
- Suboptimal performance & performance portability
  - Users cannot properly exploit the hardware / storage landscape
  - Tuning for file formats and file systems necessary at the *application* level

Manual Tuning is Non-Trivial Even for Trivial I/O

University of Reading

- Goal: Identify good settings for I/O
- Measured on Mistral Phase1; Lustre
- Run: IOR, indep. files, 10 MiB blocks
  - Proc. per node: 1,2,4,6,8,12,16
  - Stripes: 1,2,4,16,116
- Note: Slowest client stalls others



Best settings for read (excerpt)

Nodes	PPN	Stripe	W1	W2	W3	R1	R2	R3	Avg. Write	Avg. Read	WNode	RNode	RPPN
1	6	1	3636	3685	1034	4448	5106	5016	2785	4857	2785	4857	809
2	6	1	6988	4055	6807	8864	9077	9585	5950	9175	2975	4587	764
10	16	2	16135	24697	17372	27717	27804	27181	19401	27567	1940	2756	172

A Typical Current Software Stack for NWP/Climate

- Domain semantics
  - XIOS writes independent variables to one file each
  - 2nd servers for performance reasons
- Issues with the Data model in NetCDF4/HDF5
  - Performant mappings to files are limited
    - Map data semantics to one "file"
    - HDF5 shared file format notorious inefficient
  - Domain metadata is treated like normal data
    - Need for higher-level databases like Mars
  - Interfaces focus on variables but lack features
    - Workflows
    - Information life cycle management

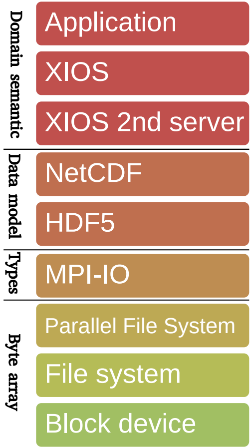
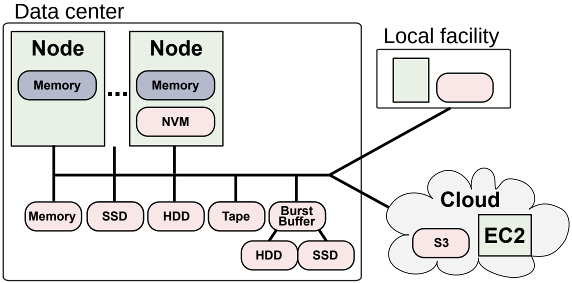


Figure: Typical I/O stack

Problem: Coexistence of Storage in Future Systems

University of Reading



- Goal: We shall be able to use all storage technologies concurrently
  - Without explicit migration etc. put data where it fits
  - Administrators just add a new technology (e.g., SSD pool) and users benefit
- Why no manual configuration, e.g., partitioning by file?
  - Reminds on implementing manual RAID across HDDs
  - Increases burden of data management

Outline

EU funded Project: ESiWACE

University of Reading

The Centre of Excellence in Simulation of Weather and Climate in Europe

- Representing the European community for
  - climate modelling and numerical weather simulation
- Goals in respect to HPC environments:
  - Improve efficiency and productivity
  - Supporting the end-to-end workflow of global Earth system modelling
  - Establish demonstrator simulations that run at highest affordable resolution
- Funding via the European Union's Horizon 2020 program (grant #675191)

http://esiwace.eu

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

Earth-System Data Middleware

University of Reading

Part of the ESiWACE Center of Excellence in H2020.

ESDM provides a transitional approach towards a vision for I/O addressing

- Scalable data management practice
- The inhomogeneous storage stack
- Suboptimal performance & performance portability
- Data conversion/merging

Earth-System Data Middleware

University of Reading

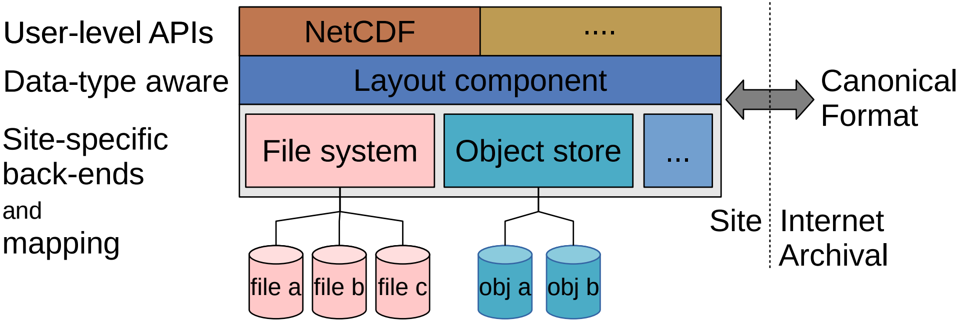
Design Goals of the Earth-System Data Middleware

1. Relaxed access semantics, tailored to scientific data generation
   - Avoid false sharing (of data blocks) in the write-path
   - Understand application data structures and scientific metadata
   - Reduce penalties of **shared** file access
2. Site-specific (optimized) data layout schemes
   - Based on site-configuration and performance model
   - Site-admin/project group defines mapping
   - Flexible mapping of data to multiple storage backends
   - Exploiting backends in the storage landscape
3. Ease of use and deployment particularly configuration
4. Enable a configurable namespace based on scientific metadata

Architecture

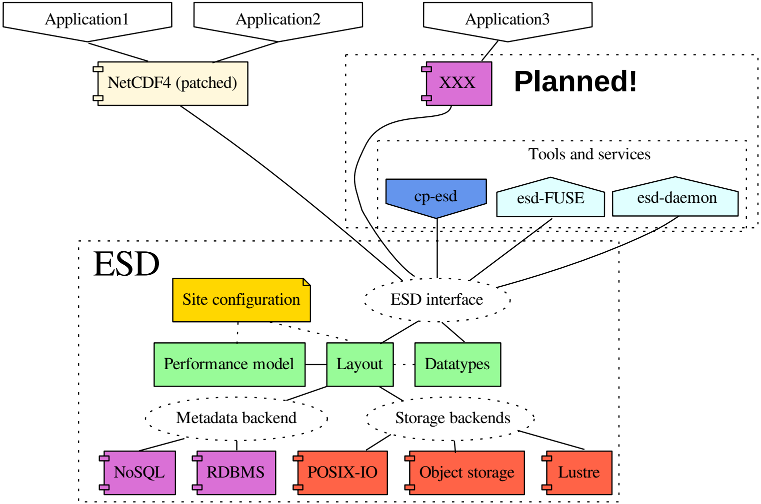
Key Concepts

- Middleware utilizes layout component to make placement decisions
- Applications work through existing API (currently: NetCDF library)
- Data is then written/read efficiently; potential for optimization inside library



Architecture: Detailed View of the Software Landscape

University of Reading



Evaluation

System

- Test system: DKRZ Mistral supercomputer
- Nodes: 200 (we have also other measurements)

Benchmark

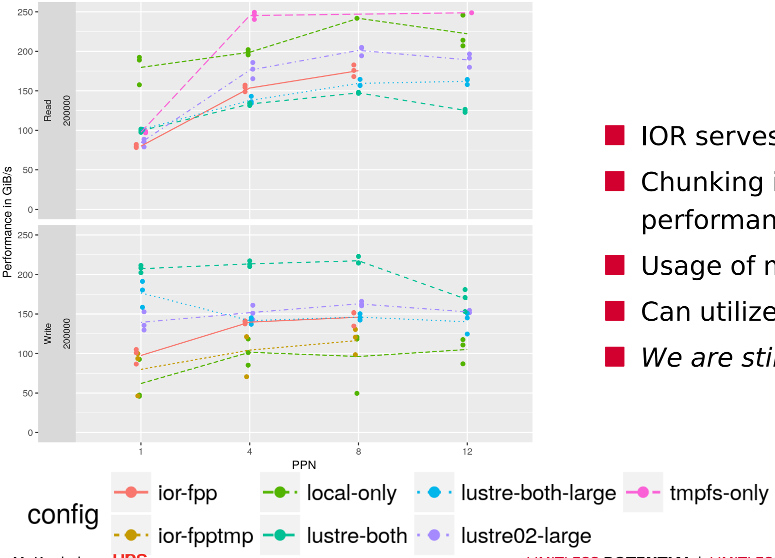
- Uses ESDM interface directly; Metadata on Lustre
- Write/read a timeseries of a 2D variable
- Grid size: $200k \cdot 200k \cdot 8Byte \cdot 10iterations$
- Data volume: size = 2980 GiB; compared to IOR performance

ESDM Configurations

- Splitting data into fragments of 100 MiB (or 500)
- Use different storage systems
- Uses 8 threads per node (max per application 400)

Measured Performance

University of Reading



- IOR serves as baseline (optimal IO)
- Chunking into files increases performance
- Usage of multiple Lustre fs +25%
- Can utilize various storage "tiers"
- *We are still working on it*

Outline

ESiWACE2 Plans for ESDM

University of Reading

ESiWACE2 follow up grant (2019-2022)

- Hardening of ESDM
- Integrate an improved performance model
- Improvements on compression (also for NetCDF)
- Optimized backends for, e.g., Clovis, IME, S3
- Integrate Workflows (Cylc) with ESDM
  - Extensions to Cylc to cover data lifecycle, I/O performance needs
  - Cylc to provide information about workflow to ESDM
  - ESDM to make superior placement decisions
- Industry proof of concepts for EDSM: Vendors to ship ESDM
- Supporting post-processing, analytics and (in-situ) visualization
  - Exploring the support of data-centric computation workflows within ESDM
  - Integration with analysis tools, e.g., Ophidia, CDO

Long Term Vision: Full Separation of Concerns

University of Reading

Decisions made by scientists

- Scientific metadata
- Declaring workflows
  - Covering data ingestion, processing, product generation and analysis
  - Data life cycle (and archive/exchange file format)
  - Constraints on: accessibility (permissions), ...
  - Expectations: completion time (interactive feedback human/system)
- Modifying workflows on the fly
- Interactive analysis, e.g., Visual Analytics
- Declaring value of data (logfile, data-product, observation)

Separation of Concerns

Programmers of models/tools

- Decide about the most appropriate API to use (e.g., NetCDF + X)
- Register compute snippets (analytics) to API
- Do not care **where** and **how** compute/store

Decisions made by the (compute/storage) system

- Where and how to store data, including file format
- Complete management of available storage space
- Performed data transformations, replication factors, storage to use
- Including scheduling of compute/storage/analysis jobs (using, e.g., ML)
- Where to run certain data-driven computations (**Fluid-computing**)
  - Client, server, in-network, cloud, your connected laptop

A Proposed Software Stack for NWP/Climate

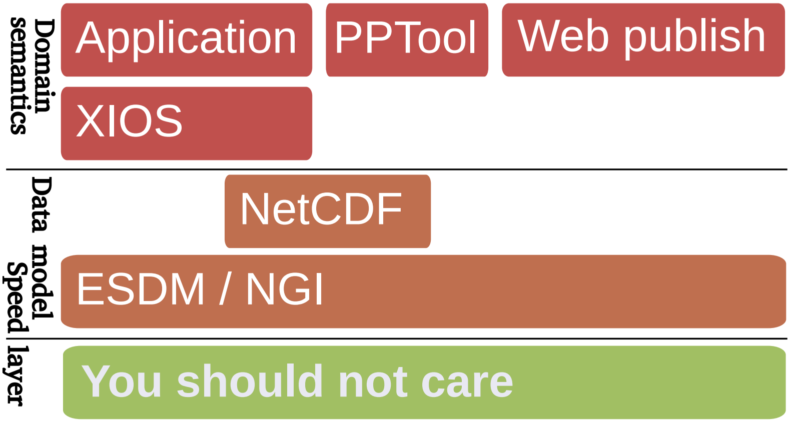


Figure: Proposed software stack

ESDM is just the Beginning: Next Generation Interfaces

University of Reading

Towards a new I/O stack considering:

- Smart hardware and software components
- Storage and compute are covered together
- User metadata and workflows as first-class citizens
- Self-aware instead of unconscious
- Improving over time (self-learning, hardware upgrades)

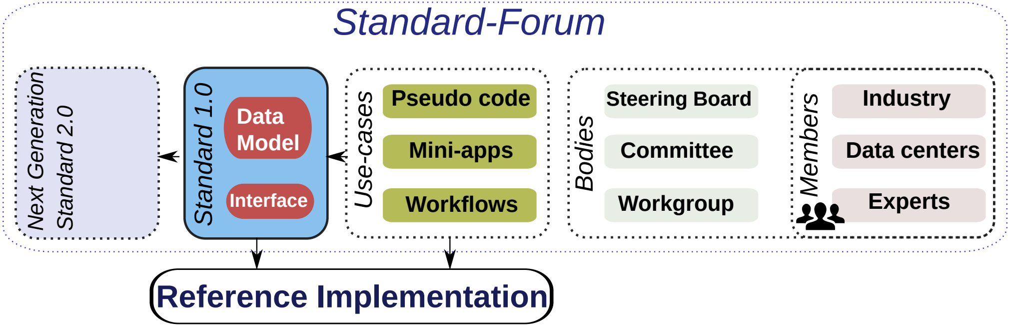
Why do we need a new domain-independent API?

- Many domains have similar issues
- It is a hard problem approached by countless approaches
- Harness RD&E effort across domains

Development of the Data Model and API

University of Reading

- Establishing a Forum (similarly to the Message Passing Interface – MPI)
- Model targets High-Performance Computing and data-intensive compute
- Open board: encourage community collaboration



Summary

- Simulation workflows in Climate and Weather are data-intensive
- Optimization requires knowledge about workflows
- Integrated and smart compute & storage is the future

Participate defining NG interfaces

- Join the mailing list / Slack
- Visit: `https://ngi.vi4io.org`

The ESiWACE project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No **675191**

esiwace

CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

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

Appendix

Scenario: Large Simulation

- Assume large scale simulation, timeseries (e.g., 1000 y climate)
- Assume manual data analysis needed (but time consuming)
- We need all 1000 y for detailed analysis!

A typical workflow execution

- Run simulation for 1000 y
  - Store various data on (online) storage
  - Keep checkpoints to allow reruns
  - Maybe backup data in archive
- Explore data to identify how to analyze data
- At some point: Run the analysis on all data
- Problem: Occupied storage capacity

Alternative Workflows Done by Scientists

Recomputation

- Run simulation
  - Store checkpoints
  - Store only selected data (wrt. resolution, section, time)
- Explore data
  - Run recomputation to create needed data (e.g., last year)
- At some point: run analysis across all data needed
- This is a manual process, must consider
  - Runtime parameters
  - System configuration/available resources
  - We are trading compute cycle vs. storage
  - It would be great if a system would consider costs...

Another Alternative Workflows

Provided by more intelligent storage and better workflows

- Run simulation
  - Store checkpoints on node-local storage
    - Redundancy: from time to time restart from another node
  - Store selected data on online storage (e.g., 1% of volume)
    - Also store high-resolution data sample (e.g., 1% of volume)
  - Store high-resolution data directly on tape
- Explore data on snapshot
- Month later: schedule analysis of data needed
  - The system retrieves data from tape
  - Performs the scheduled operations on streams while data is pulled in
  - Informs user about analysis progress
- Some people do this manually or use some tools to achieve similarly
  - Aim for domain & platform independence and heterogenous HPC landscapes

Scenario: Data Organization

Goal: Semantic Namespace

- Provide features of data repositories (e.g., MARS) to explore data
- User-defined properties but provide means to validate schemas
- Similar to MP3 library ...

High-Level questions addressed by them

- What experiments did I run yesterday?
- Show me the data of experiment X, with parameters Z...
- Cleanup unneeded temporary stuff from experiment X
- Compare the mean temperature of one model for one experiment across model versions