Progress of WP4: Data at Scale

WP4 Team

ESiWACE GA

September 2021

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

Reminder: WP4: Data Systems at Scale

Objectives

To mitigate the effects of the data deluge from high-resolution simulations (project objective d) by

1. Supporting **data reduction in ensembles** by providing tools to carry out ensemble statistics “in-flight” and compress ensemble members
2. **Hiding complexity** of multiple-storage tiers (middleware between NetCDF and storage) with industrial prototype backends
3. Delivering **portable workflow support** for manual migration of semantically important content between disk, tape, and object stores

⇒ *Ensemble tools, storage middleware, storage workflow*

Outline

1 T2: Ensemble Services

2 T4: Semantic Storage

3 T3: ESDM

4 T5: Workflows

5 T7: Industry PoC

Outline

1 T2: Ensemble Services

2 T4: Semantic Storage

3 T3: ESDM

4 T5: Workflows

5 T7: Industry PoC

T2: Ensemble Services (Lister, Cole, Wilson)

Reminder: Goals

- Run ensemble members in parallel and do diagnostics “on-the-fly” using XIOS.
    - e.g., store mean and variance of ensemble results (instead of all members)

Three Key Activities

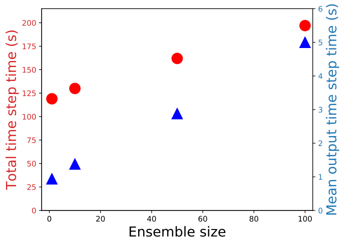
- Implement XIOS ensembles (In the Unified Model Atmosphere) on one MPI communicator.
- Proceed to real science demonstrator (with Met Office in WP1).
- Handle the risk of an ensemble member failure.

Future work: EXCALIBUR

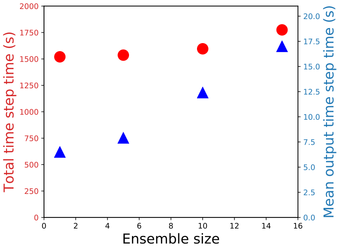
- Do this for coupled models including NEMO.

Ensemble Services: Improved Scaling

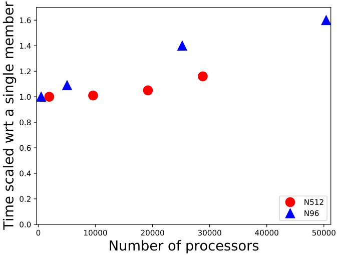
N96 (192x144x85)



N512 (1024x768x85)



Scaling



Experiments with one ensemble diagnostic and no normal output

- Relatively poor scaling for CMIP type resolutions
- Much better scaling at higher resolution.
- Ensemble calculations will be lost in noise with (some) normal I/O.

Ensemble Services: Next Steps

Further Modifications to the UM

Happening now

- Moving to all output via XIOS into NetCDF (many edge cases)
- Configuring required output.
  - Internal model pipework to route to XIOS
  - XMLApp so user can configure outputs.
- Compression via Gaussian Grids (optional, done)
- Upgrading XIOS versions.
- Performance profiling and tuning.

Further Modifications to Suite Control

- Managing an ensemble.
- Managing error handling (next slide)
- Managing data migration (JDMA and JASMIN, later slides)

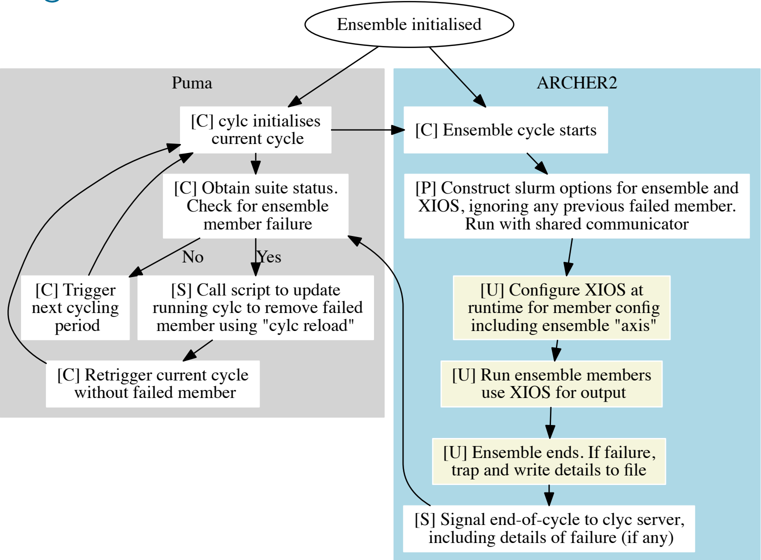
Towards Science runs for WP1

- What experiments & resolutions?
- Developing appropriate ensemble diagnostics.
- What output do we need from ensemble members?

Ensemble Services: Handling a dead ensemble member

MPI Failure?

- Original Goal: Trap MPI error from failing member.
- Better solution: Handle it in cylc.
- Algorithm and example of usage in D4.1


Ensemble initialised

Puma

[C] cylc initialises current cycle

[C] Obtain suite status. Check for ensemble member failure

No

Yes

[C] Trigger next cycling period

[S] Call script to update running cylc to remove failed member using "cylc reload"

[C] Retrigger current cycle without failed member

ARCHER2

[C] Ensemble cycle starts

[P] Construct slurm options for ensemble and XIOS, ignoring any previous failed member. Run with shared communicator

[U] Configure XIOS at runtime for member config including ensemble "axis"

[U] Run ensemble members use XIOS for output

[U] Ensemble ends. If failure, trap and write details to file

[S] Signal end-of-cycle to clyc server, including details of failure (if any)

Outline

1 T2: Ensemble Services

2 T4: Semantic Storage

3 T3: ESDM

4 T5: Workflows

5 T7: Industry PoC

T4: Semantic Storage (Massey); the story thus far (early 2021)

Joint Data Migration App

- Aim: To manage large migrations between disk and tape or disk and object-store (and vice-versa).
- Status: In production on JASMIN. o(1PB) of data held by users.
- Issues: Users positive about functionality but not performance, particularly to tape. Probably too much (repeated) verification. No real semantics, still need to retrieve a file to know what as in it.

S3NetCDF (Python Module)

- Aim: S3 aware replacement for netcdf4-python.
- Status: At V2 utilising notions of an "aggregation file" and "fragments". The former on POSIX disk, the latter anywhere (but in particular, behind and S3 interface).
- Issues: Some use cases overtaken by zarr and netcdf c-lib. Performance issues. Aggregation rules & syntax not widely known & supported.

Decision Time

- Choices: New Excalibur Funding Available, so there was a clear route to continued funding, but when and how should we take-on the lessons learned. Now, or between projects? But projects overlapped?

Decision Time

- Choices: New Excalibur Funding Available, so there was a clear route to continued funding, but when and how should we take-on the lessons learned. Now, or between projects? But projects overlapped?

- Decisions:
  - Take the planned JDMA refactor, but build into a bigger activity.
  - Take the best ideas of S3NetCDF (smart aggregation) and "socialise" them.
  - Take the existing software of both and refactor into even further modularity so wider chance of re-use of both end-to-end functionality and components.

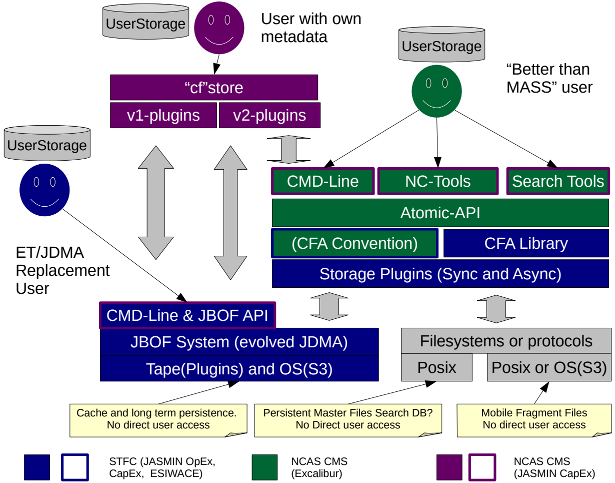
Three distinct tape and object store use cases

Use Cases

Multiple funding sources:

1. Managing "Just a bunch of files" (evolution of JDMA).
2. Adding semantic information.
3. Portable "MASS" for NetCDF functionality.

(Not including Object-Store only use case, e.g. pangeo)



UserStorage

User with own metadata

UserStorage

"Better than MASS" user

"cf"store

v1-plugins

v2-plugins

UserStorage

CMD-Line

NC-Tools

Search Tools

Atomic-API

(CFA Convention)

CFA Library

Storage Plugins (Sync and Async)

ET/JDMA Replacement User

CMD-Line & JBOF API

JBOF System (evolved JDMA)

Tape(Plugins) and OS(S3)

Filesystems or protocols

Posix

Posix or OS(S3)

Cache and long term persistence.
No direct user access

Persistent Master Files Search DB?
No Direct user access

Mobile Fragment Files
No direct user access

■ □ STFC (JASMIN OpEx, CapEx, ESIWACE)

■ NCAS CMS (Excalibur)

■ □ NCAS CMS (JASMIN CapEx)

Tape: Developments and Progress

- New name: NLDS (Near Line Data Store)
- Key idea: move to using object store as a cache for a "lightweight" HSM.
- Open Policy Agent for “HSM” policies.
- Using RabbitMQ to manage work queue.
- OAuth2 for authorisation.
- CERN’s FT3 to manage transfers.
    - S3, CTA, and maybe StrongBox (?) plugins?
- New staff member (Jack Leland) joined Neil Massey at STFC to work on it.

Meanwhile:

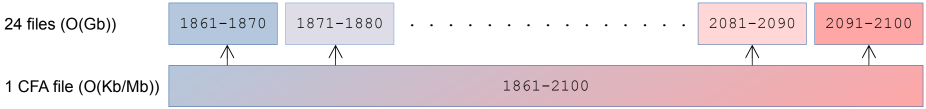
- Three member ten-year high-resolution N1280 (10km) ensemble begun on ARCHER2, is using JDMA in Cylc suite running on ARCHER2 to write to JASMIN.
- (So backwards compatible with JDMA is necessary.)

Aggregation File Syntax

`https://github.com/NCAS-CMS/cfa-conventions`

- Storing aggregations of existing datasets is useful
  - Data analysis
  - Archive curation

- **Example:** For a timeseries of surface air temperature from 1861 to 2100 that is stored in 24 files each spanning 10 years, it is useful to view this as if it were a single dataset spanning 240 years.



24 files (O(Gb))

1861-1870 1871-1880 . . . . 2081-2090 2091-2100

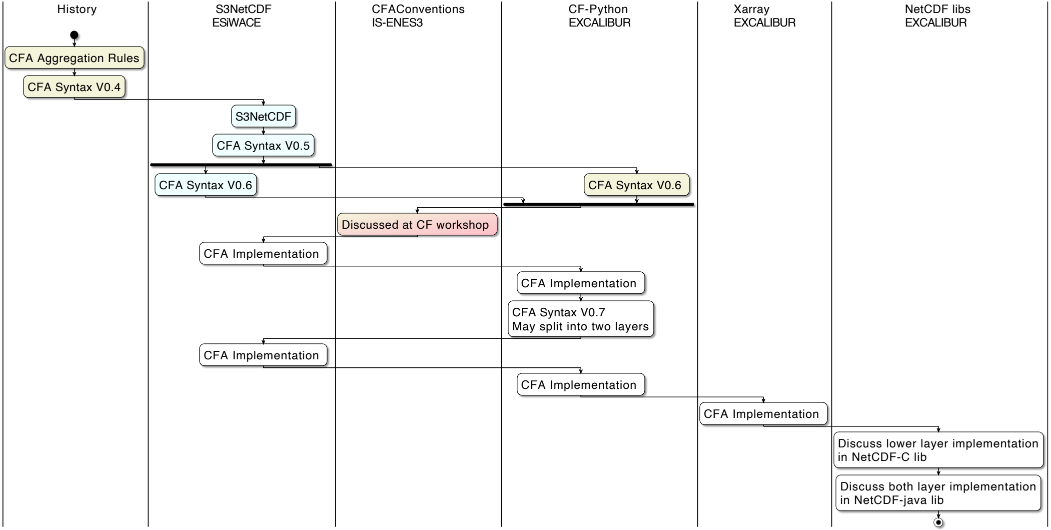
1 CFA file (O(Kb/Mb))

1861-2100

- **CFA-netCDF** is a(nother) proposed standard for recording an aggregation without copying the data so that it doesn't need to be remade on-the-fly (expensive), and is available as an archive index

David Hassell and Neil Massey

Aggregation Files: where next?



History

S3NetCDF
ESiWACE

CFAConventions
IS-ENES3

CF-Python
EXCALIBUR

Xarray
EXCALIBUR

NetCDF libs
EXCALIBUR

CFA Aggregation Rules

CFA Syntax V0.4

S3NetCDF

CFA Syntax V0.5

CFA Syntax V0.6

CFA Syntax V0.6

Discussed at CF workshop

CFA Implementation

CFA Implementation

CFA Syntax V0.7
May split into two layers

CFA Implementation

CFA Implementation

CFA Implementation

Discuss lower layer implementation
in NetCDF-C lib

Discuss both layer implementation
in NetCDF-java lib

Outline

1 T2: Ensemble Services

2 T4: Semantic Storage

3 T3: ESDM

4 T5: Workflows

5 T7: Industry PoC

Reminder: Earth-System Data Middleware (ESDM)

A transitional approach towards a vision for I/O addressing

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

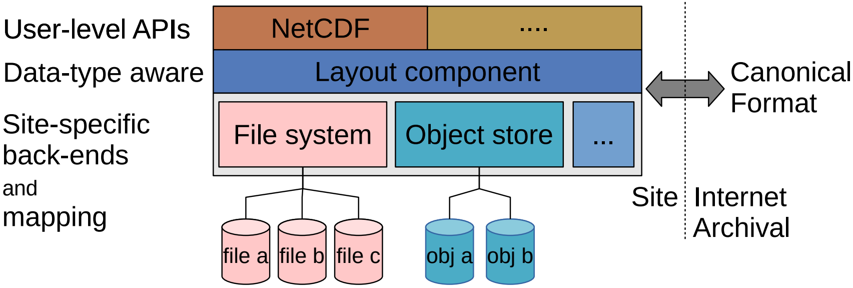
Design goals of the Earth-System Data Middleware

1. Relaxed access semantics, tailored to scientific data generation
2. Site-specific (optimized) data layout schemes
3. Ease of use and deploy a particular configuration
4. (Enable a configurable namespace based on scientific metadata)

Reminder: Architecture

Key concept: Decouple data localization decisions from science

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



Reminder: ESDM as NetCDF Drop-In is Easy to Use

- Create a ESDM configuration with storage locations
- Run esdm-mkfs to prepare storage systems (e.g., mkdir on POSIX)
- Change file names when running NetCDF applications
  - The namespace of ESDM is separated from the file system (hierarchical too)
  - NetCDF can use ESDM by just utilizing the **esdm://** prefix
- Examples:
  - Import/Inspection/Export of data using NetCDF

    ```
    $ nccopy test_echam_spectral.nc esdm://user/test_echam_spectral
    $ ncdump -h esdm://user/test_echam_spectral
    $ nccopy -4 esdm://user/test_echam_spectral out.nc
    ```

  - Usage in XIOS, change iodef. Example:

    ```
    <file id="output" name="esdm://output" enabled=".TRUE.">
    ```

    prec=8 in axis_definition, domain_definition and field_definition

Selected Activities: Status Overview

- Submission of WP4 deliverable
- Integrated ESDM with Paraview, patch for CDO support
- ESDM NetCDF supported version updated to current NetCDF Git
- Benchmarking efforts at CMCC and NCAS
- S3 backend implemented
- Prototype for transparent data transformation/replication upon reads
- Ophidia integration / evaluation (details next slide)

Integration of Ophidia with ESDM and Evaluation

- Different integration strategies implemented
  - Linking Ophidia with the ESDM-NetCDF library
    - Code rebuilding and minor modifications required
  - Direct integration of the ESDM API in Ophidia
    - New Ophidia operators for data loading and storing developed (OPH_IMPORTESDM, OPH_EXPORTESDM)
- Preliminary testing of the two integrations performed
  - Initial (small scale) results show no clear difference in the two approaches (direct integration slightly faster in some cases)
  - More extensive benchmarking is needed (planned for Y4)
- Discussion between WP4 and WP5 for Ophidia extensions for in-flight analytics based on ESDM
  - Use/testing of active-storage solutions to be evaluated

ESiWACE2 TODOs for ESDM

- Evaluation of ESiWACE-relevant scenarios
  - Pending activity to explore OpenIFS or NEMO at the GWDG
- Industry proof of concepts for EDSM, i.e., shipping of HW with software
- WP5: Supporting post-processing, analytics and (in-situ) visualization
- Optional
  - Hardening and optimization of ESDM
    - Integrate improved performance model
    - Further backend optimization
  - Features
    - Complete replicate data upon read (adaptive fragments) - publication was pending
    - Evaluation of structured (chunked) vs. flexible (ESDM) fragments - pub was pending
    - NoSQL metadata backend

Outline

1 T2: Ensemble Services

2 T4: Semantic Storage

3 T3: ESDM

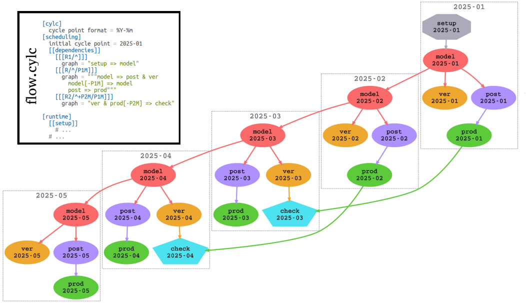
4 T5: Workflows

5 T7: Industry PoC

Reminder: T5: Workflows

- Goal: Explore higher-level abstraction - scientists don't need to worry where data is
- Data placement could be optimized by considering available hardware
  - Different and heterogenous storage systems available
  - Prefetching of data, using local storage, using IME hints, ...
- Status: We created a design document in the consortium

- A workflow consists of many steps
  - Repeated for simulation time
  - E.g., weather for 14 days
- Cylc workflow specifies
  - Tasks with commands
  - Environment variables
  - Dependencies



flow.cylc

```
[cylc]
  cycle point format = %Y-%m
[scheduling]
  initial cycle point = 2025-01
  [[dependencies]]
    [[[R1/^]]]
      graph = "setup => model"
    [[[R/^/P1M]]]
      graph = """model => post & ver
        model[-P1M] => model
        post => prod"""
    [[[R2/^+P2M/P1M]]]
      graph = "ver & prod[-P2M] => check"

[runtime]
  [[setup]]
    # ...
  # ...
```

Activities

- Performed first analysis of integration between Cylc and ESDM
- Plan is not to pursue this research task further
  - Not a problem for our demonstrator
  - Could harvest some low-hanging fruits if there would be high interest in ESiWACE
- Action: DDN (Konstantionos) will document IME SLURM integration

Outline

1 T2: Ensemble Services

2 T4: Semantic Storage

3 T3: ESDM

4 T5: Workflows

5 T7: Industry PoC

T7: Industry PoC

- Goal: Usage of ESDM in a data center storage environment, using either Vendor storage appliance or Vendor deployment of storage software on COTS hardware
  - DDN to focus on providing a prototype appliance package
  - Seagate to focus on deploying Motr/Mero environment in weather/climate center
  - Motr is now fully open source and should work with COTS hardware
    - DKRZ identified as potential site for Motr deployment
    - Plan to explore aspects such as performance and function shipping