WP4: Highlights, challenges and outlook

Julian Kunkel\textsuperscript{1,7}, Bryan N. Lawrence\textsuperscript{2,3}, Jakob Luettgau\textsuperscript{7}, Neil Massey\textsuperscript{4}, Alessandro Danca\textsuperscript{5}, Sandro Fiore\textsuperscript{5}, Huang Hu\textsuperscript{6}

\textsuperscript{1}Department of Computer Science, University of Reading
\textsuperscript{2}UK National Centre for Atmospheric Science
\textsuperscript{3}Department of Meteorology, University of Reading
\textsuperscript{4}STFC Rutherford Appleton Laboratory
\textsuperscript{5}CMCC Foundation
\textsuperscript{6}Seagate Technology LLC
\textsuperscript{7}DKRZ

11 March 2019
Outline

1. Introduction

2. Task 1: Business

3. Task 2: ESDM


5. Summary & Next Steps

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
Project Organisation

WP1 Governance and engagement

WP2 High-resolution demonstrators

WP3 Usability

WP4 Exploitability

- Exploiting high volume data: How to get more science done
- Storage layout for Earth system data
- Methods of exploiting tape

WP5 Management and Dissemination
Work Package 4 — Exploitability (of data); Overview

Partners

DKRZ, STFC, CMCC, Seagate, UREAD

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

Task 4.1

Business Models

- Documentation
  - Coarse-grained model
  - Fine-grained model
- D4.1

Task 4.2

New Storage Layout

- Software & Design
  - ESD Middleware
- Design delivered D4.2
- Initial benchmarks
- Development ongoing

Task 4.3

New Tape Methods

- Software
  - JDMA data migration
- Prototype in place
- D4.4; Wrapup ongoing
Outline

1. Introduction
2. Task 1: Business
3. Task 2: ESDM
5. Summary & Next Steps
Coarse-Grained Models

Simple graph models

High-level representation
- Hardware/software
- Purpose: Ease understanding

Includes:
- performance
- resilience
- cost

Deliverable D4.1 (done)

Scenarios discussing architectural changes for data centres, and implications for cost/performance
Some Examples of Business Considerations

One cost model of storage based on DKRZ

- Tape: 12 € per TB/year
- Software licenses for tape are driving the costs!
- Parallel Disk: 28 € TB/year
- Object storage: 12.5 € TB/year (without software license costs)
- Cloud: $ 48 TB/year (only storage, access adds costs)
- Alternative models: 8 € / 153 € for tape/disk per year
- Idle (unused) data is an important cost driver!

Lüttgau, Kunkel; Cost and Performance Modeling for Earth System Data Management and Beyond; High-Performance Computing; ISC-HPC workshops
Fine-Grained Performance Modelling

Detailed Modelling

A simulator has been developed; covers
- HW, 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 – particularly tape

Lüttgau, Kunkel; Simulation of Hierarchical Storage Systems for TCO and QoS; High-Performance Computing; ISC-HPC workshops
### Challenges

- Costs for hardware/software often intertwined, hard to disentangle
- Obscured behavior of hardware/software (e.g., HPSS)
- We had only a small budget to address these issues

### Outlook

- Modelling and simulation remains important
  - How can we best use heterogeneous systems?
- No continuation of activity in ESiWACE 2 (but we’ll continue outside)
Outline

1. Introduction
2. Task 1: Business
3. Task 2: ESDM
5. Summary & Next Steps
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

3. Ease of use and deployment particularly configuration
Benefits

■ Independent, share-nothing lock-free writes from parallel applications
■ Storage layout is optimized to local storage
  ▶ Exploits characteristics of diverse storage
  ▶ Preserve compatibility by creating platform-independent file formats on the site boundary/archive
■ Less performance tuning from users needed
  ▶ One data structure can be fully or partially replicated with different layouts
  ▶ Using multiple storage systems concurrently
■ (Expose/access the same data via different APIs\(^1\))
■ (Flexible and automatic namespace\(^1\))

\(^1\)Explored outside the ESiWACE scope
Key Concepts

- Middleware utilizes layout component to make placement decisions
- Applications work through existing API
- Data is then written/read efficiently; potential for optimization inside library
Evaluation of the Prototype at DKRZ Mistral

**System**
- Test system: DKRZ Mistral supercomputer
- Nodes: 200

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

**ESDM Configurations**
- Splitting data into fragments of 100 MiB (or 500)
- Use different storage systems
Measured Performance

- Read is handled inefficiently in ESDM at the moment. We will optimize the read path.

Kunkel & Lawrence (WP4 Team)

WP4: Highlights, challenges and outlook

11 March 2019
Data Backends – DDN Object Store (CMCC)

WOS Prototype

- Backend works
- Developed C wrapper for the C++ DDN WOS libraries
- Designed a parallel approach for independent / multiple write operations on WOS storage
Deployment Testing Example

Test and Deployment

Ophidia (in-memory data analytics) as a test application for ESDM

- **Import and Export**
  - Ophidia operators adapted for integration with ESDM storage
  - Uses patched NetCDF

- **ESDM successfully built on:**
  - Athena HPC Cluster
  - OphidiaLab

- Creation of a VM for the whole software stack

- **Deployment Testing Example**
Architecture: View of the Software Landscape as Planned

- Application1
- Application2
  - NetCDF4 (patched)
- HDF5 VOL (unmodified)
- ESD (Plugin)
- ESD (Plugin)
- Application3
  - GRIB
- cp-esd
- esd-FUSE
- esd-daemon

ESD interface

- Site configuration
- Performance model
- Layout
- Datatypes
- Metadata backend
- Storage backends
  - NoSQL
  - RDBMS
  - POSIX-IO
  - Object storage
  - Lustre

Tools and services

Kunkel & Lawrence (WP4 Team)
ESDM Development

Status

- ESDM Architecture Design for Prototype (D4.2)
- Multi-threaded data path
- Data backend Plugins for POSIX, CLOVIS, WOS ( Reached: MS7)
- Trivial POSIX metadata store on the shared file system
- Proof of concept for adaptive tier selection in HDF5
  - But only for a trivial use case!
- 60%: ESDM library implementation
- Partial implementation for HDF5 VOL
- Evaluation of **ESDM benchmark** at DKRZ, STFC, CMCC ( Reached: MS9)
- Started direct NetCDF integration – prototype for the write-path works

₂Note that for execution of applications not all 100% functionality will ever be needed.
## Challenges & Outlook

### Challenges

- Choosing HDF5 (VOL) wasted too much of effort
- Backend: DDN discontinued WOS
- Core-development with too few FTE for PostDoc
- People leaving teams (Seagate, DKRZ)
- Teamwork between DKRZ and Seagate was suboptimal
- Identification of NoSQL Metadata backend
Outlook

- Building a performance model for WOS/CLOVIS as blueprint for backends
- Hired a PostDoc at UoR to continue effort
- Goal: Supporting a subset of NetCDF applications
  - NetCDF benchmark
  - Toy model: Shallow water equation
  - Ophidia: use it in one big data workflow
- Improve data plugin for POSIX
- Optimize read path exploring a NoSQL backend
- Run small benchmarks at sites
  - CLOVIS performance in various configurations on a reasonable cluster
<table>
<thead>
<tr>
<th></th>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Task 1: Business</td>
</tr>
<tr>
<td>3</td>
<td>Task 2: ESDM</td>
</tr>
<tr>
<td>4</td>
<td>Task 3: New Tape Methods</td>
</tr>
<tr>
<td>5</td>
<td>Summary &amp; Next Steps</td>
</tr>
</tbody>
</table>
Approach

Semantic Storage Library

Task 3: Developing new tape access strategies and software . . . higher bandwidth to tape storage and increased storage redundancy.

- **Increase bandwidth to/from tape by exploiting RAID-to-TAPE.**
  - Decided that this was too difficult to do in a portable manner and that portable (tape + object store) workflow was a more important initial priority.

- Provide a portable library to address user management of data files on disk (POSIX and/or Object Store) and tape which
  1. does not *require* significant sysadmin interaction, but
  2. can make use of local customisation if available/possible
  3. exploits existing metadata conventions
  4. prototype can be deployed fast enough that we can use it for Exascale Demonstrator
Architecture

Two Key Components

1. S3NetCDF — replacement for NetCDF4-python with support for object stores
2. CacheFace — a portable frontend for managing content in object stores/tape
Architecture

Two Key Components

1. S3NetCDF — replacement for NetCDF4-python with support for object stores
2. CacheFace — a portable frontend for managing content in object stores/tape

Information Structure

Exploiting the Climate Forecast Aggregation (CFA) Framework\(^1\), which

1. Defines how CF fields may be combined into one larger field
2. Is fully general and based purely on CF metadata
3. Includes a syntax for storing an aggregation in a NetCDF file using JSON string content to point at aggregated files

\(^1:\)https://goo.gl/DdxGtw
S3NetCDF (working title)

File split following CFA conventions

Object
Store
Each object is a valid NetCDF File

App
S3netCDF
Local cache

Architecture

- Master Array File is a NetCDF file containing dimensions and metadata for the variables including URLs to fragment file locations
- Master Array file optionally in persistent memory or online, nearline, etc. NetCDF tools can query file CF metadata content without fetching them
**S3NetCDF (working title)**

File split following CFA conventions

---

**Status:**

- Prototype released (milestone 7B). Subsequent refactoring complete (October 2018) in preparation for parallelisation.
- ESiWACE1 goal: add prototype parallelisation, measure performance, publish paper and more complete usage documentation. (ESiWACE2: performance, integrate components with ESDM).
JDMA: a Prototype Tape Library for Advanced Tape Subsystems

- **JDMA: Joint Data Migration App(lication)**
- **A general-purpose multi-tiered storage library**
  - Provides a single API to users to move data to and from different systems
  - HTTP API running on webserver, database records requests and file metadata
  - Command line client which interfaces to HTTP API
- **Multiple storage “backends” supported via plugin**
  - Amazon S3 (Simple Storage Solution) for Object Stores and AWS
  - FTP, also for tape systems with a FTP interface
  - Elastic Tape – a proprietary tape system based on CASTOR
- **A number of daemons (scheduled processes) carry out the data transfer**
  - Asynchronously
  - On behalf of the user
JDMA System Architecture

- Backends:
  - ElasticTape
  - FTP
  - ObjectStore

- Worker Processes:
  - jdma_lock
  - jdma_pack
  - jdma_transfer
  - jdma_monitor
  - jdma_verify
  - jdma_tidy

- Database

- JDMA client

- Web server

- HTTP API
Outline

1 Introduction
2 Task1: Business
3 Task 2: ESDM
4 Task 3: New Tape Methods
5 Summary & Next Steps
### Summary

#### Software

1. **ESDM**: Performance-portable I/O with NetCDF on heterogeneous storage
2. **S3NetCDF**: Prototype for handling object store/tape
3. **JDMA**: portable, lightweight (towards HSM) system

#### ESiWACE1 Goals

1. **ESDM**: Extend usability, complete NetCDF integration, improve plugin, layout, and performance
2. **S3NetCDF** – parallelise and publicises. Release prototype complete system.
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
The PTA multi-model workflow implemented in Ophidia has been executed and validated at CMCC on 11 models from CMIP5 experiment for a total of 181 tasks, 2.5 minutes, 96 cores on OphidiaLab