Efficient Management of Self-Describing Data Formats

Kira Duwe

kira.duwe@epfl.ch

May 8, 2023

DIAS
(Data-Intensive Applications and Systems Lab)
École polytechnique fédérale de Lausanne (EPFL)
Lausanne, Switzerland

Acknowledgments

- Work was done at
  - University of Hamburg (UHH)
  - Otto-von-Guericke University (OVGU) Magdeburg

    https://www.parcio.ovgu.de/
- Thesis was supervised by Michael Kuhn
- Funded by DFG (German Research Foundation) - 417705296
- More information about the CoSEMoS (Coupled Storage System for Efficient Management of Self-Describing Data Formats) project can be found at https://cosemos.de.

Outline

Current Problems in HPC Storage Systems

HPC: High-Performance Computing

I/O: Input/Output

Current Problems in HPC Storage Systems

- Growing data sizes

HPC: High-Performance Computing

I/O: Input/Output

Current Problems in HPC Storage Systems

- Growing data sizes
- Hardware hierarchy

HPC: High-Performance Computing

I/O: Input/Output

Current Problems in HPC Storage Systems

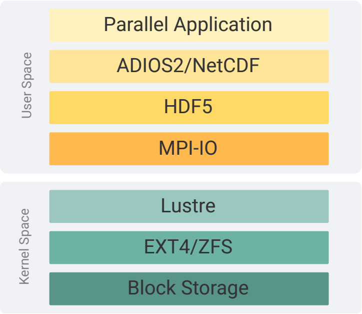
- Growing data sizes
- Hardware hierarchy
- Large software stack

HPC: High-Performance Computing

I/O: Input/Output

Current Problems in HPC Storage Systems

- Growing data sizes
- Hardware hierarchy
- Large software stack



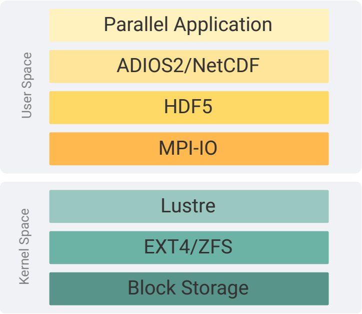
HPC I/O Software stack

HPC: High-Performance Computing

I/O: Input/Output

Current Problems in HPC Storage Systems

- Growing data sizes
- Hardware hierarchy
- Large software stack
- Optimisation for different goals



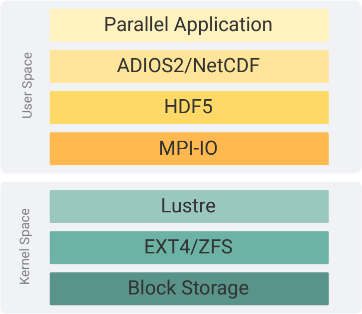
HPC I/O Software stack

HPC: High-Performance Computing

I/O: Input/Output

Current Problems in HPC Storage Systems

- Growing data sizes
- Hardware hierarchy
- Large software stack
- Optimisation for different goals
- No application changes if possible



HPC I/O Software stack

HPC: High-Performance Computing

I/O: Input/Output

BP: binary packed

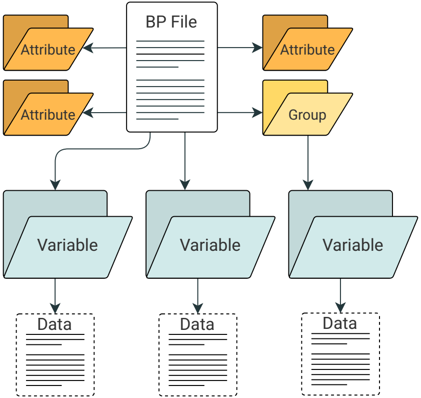
I/O Libraries and their Data Formats

- Common libraries: HDF5, NetCDF, ADIOS2
- Self-explanatory data through annotation
- Ease of data sharing

BP: binary packed

I/O Libraries and their Data Formats

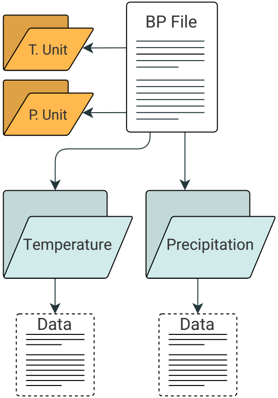
- Common libraries: HDF5, NetCDF, ADIOS2
- Self-explanatory data through annotation
- Ease of data sharing



BP: binary packed

I/O Libraries and their Data Formats

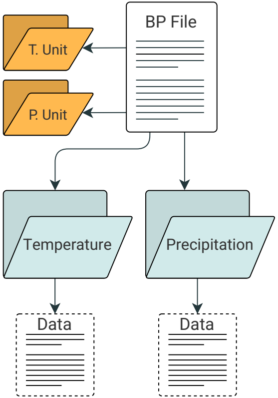
- Common libraries: HDF5, NetCDF, ADIOS2
- Self-explanatory data through annotation
- Ease of data sharing



BP: binary packed

I/O Libraries and their Data Formats

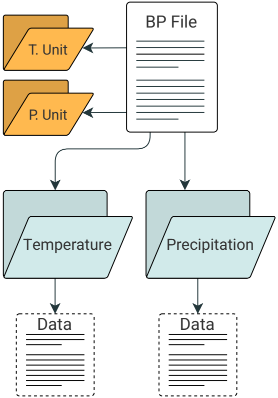
- Common libraries: HDF5, NetCDF, ADIOS2
- Self-explanatory data through annotation
- Ease of data sharing
- Data characteristics, e.g. minima/maxima



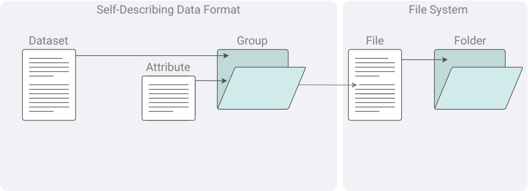
BP: binary packed

I/O Libraries and their Data Formats

- Common libraries: HDF5, NetCDF, ADIOS2
- Self-explanatory data through annotation
- Ease of data sharing
- Data characteristics, e.g. minima/maxima
- Block x = data written by process x



BP: binary packed

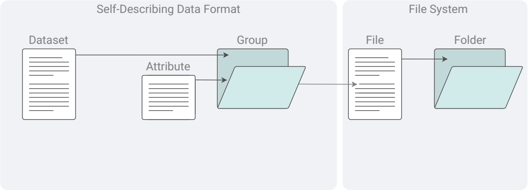


FS: file system

SDDF: self-describing data format

Characterisation of Data Categories

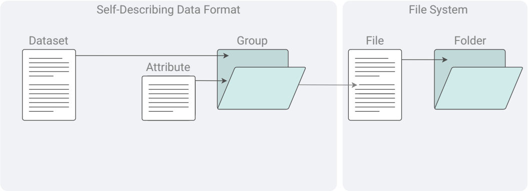
- Currently: FS metadata and data



FS: file system
SDDF: self-describing data format

Characterisation of Data Categories

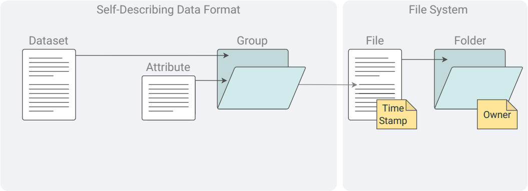
- Currently: FS metadata and data



FS: file system
SDDF: self-describing data format

Characterisation of Data Categories

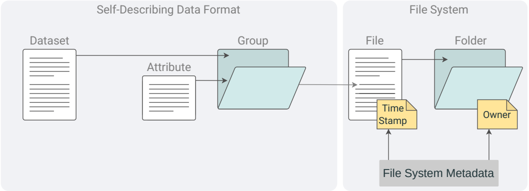
- Currently: FS metadata and data



FS: file system
SDDF: self-describing data format

Characterisation of Data Categories

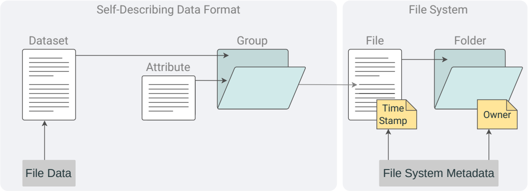
- Currently: FS metadata and data



FS: file system
SDDF: self-describing data format

Characterisation of Data Categories

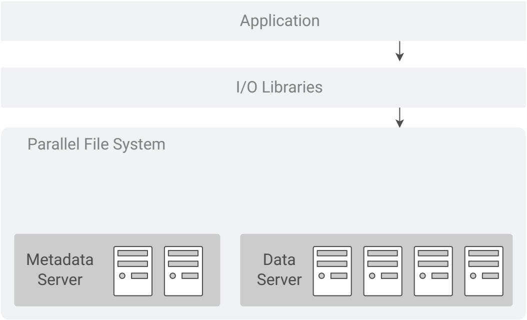
- Currently: FS metadata and data



FS: file system
SDDF: self-describing data format

Characterisation of Data Categories

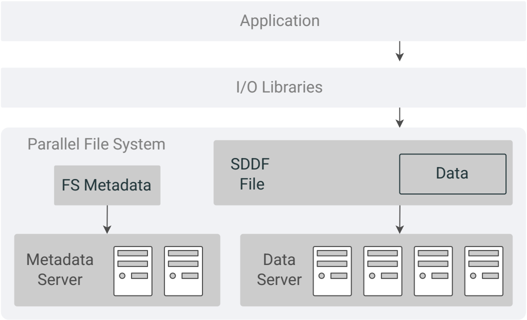
- Currently: FS metadata and data
- Metadata and data server



FS: file system
SDDF: self-describing data format

Characterisation of Data Categories

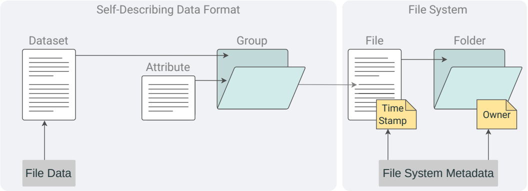
- Currently: FS metadata and data
- Metadata and data server



FS: file system
SDDF: self-describing data format

Characterisation of Data Categories

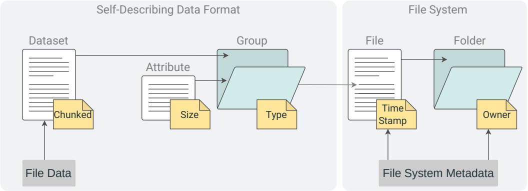
- Currently: FS metadata and data
- Metadata and data server
- Additional information in SDDF files



FS: file system
SDDF: self-describing data format

Characterisation of Data Categories

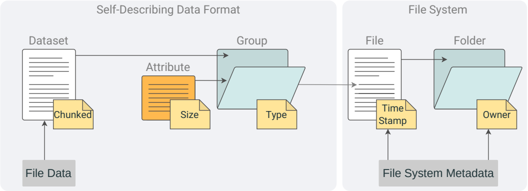
- Currently: FS metadata and data
- Metadata and data server
- Additional information in SDDF files



FS: file system
SDDF: self-describing data format

Characterisation of Data Categories

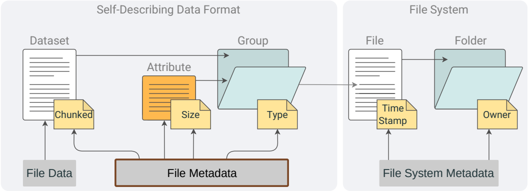
- Currently: FS metadata and data
- Metadata and data server
- Additional information in SDDF files



FS: file system
SDDF: self-describing data format

Characterisation of Data Categories

- Currently: FS metadata and data
- Metadata and data server
- Additional information in SDDF files



File Metadata

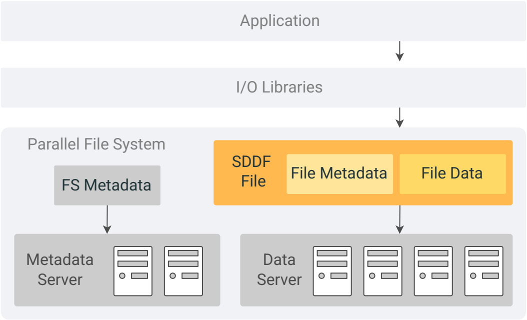
FS: file system
SDDF: self-describing data format

Characterisation of Data Categories

- Currently: FS metadata and data
- Metadata and data server
- Additional information in SDDF files

File Metadata

- Stored on data servers

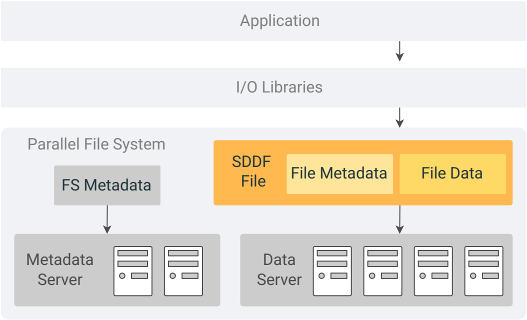


Current Management

FS: file system
SDDF: self-describing data format

Characterisation of Data Categories

- Currently: FS metadata and data
- Metadata and data server
- Additional information in SDDF files



Current Management

File Metadata

- Stored on data servers
- Access too limited

FS: file system
SDDF: self-describing data format

Outline

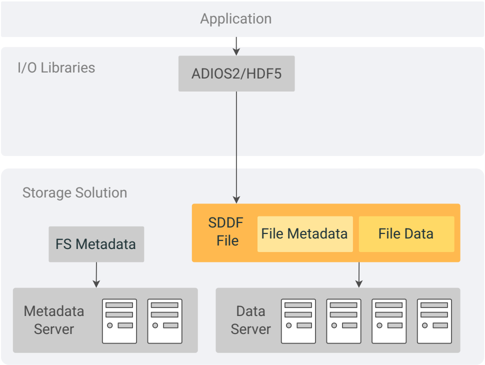
Motivation

Design

Evaluation

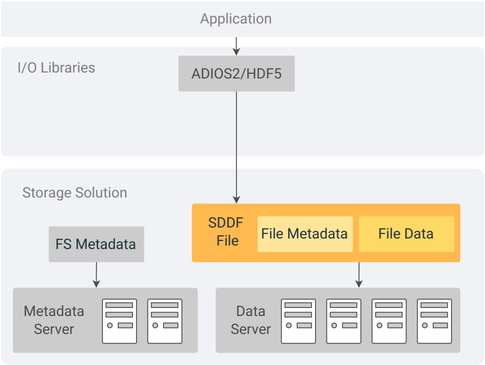
Transparent Format Dissection

Transparent Format Dissection

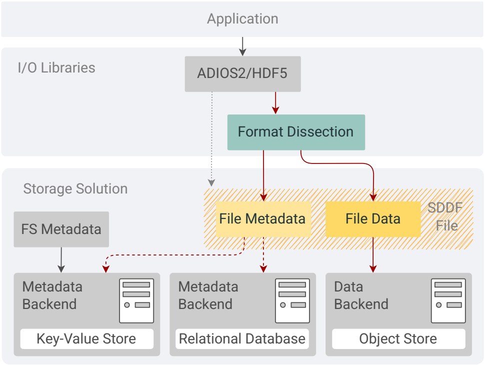


Currently

Transparent Format Dissection

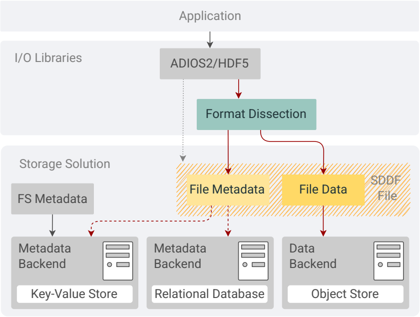


Currently



Proposal

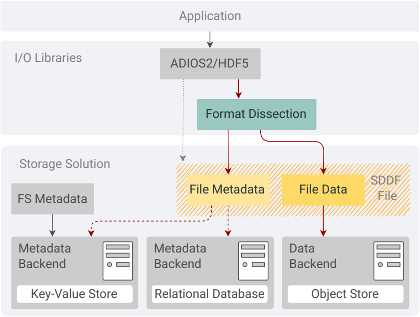
Implementing Format Dissection



Format Dissection

Implementing Format Dissection

→ Map file metadata to storage backends

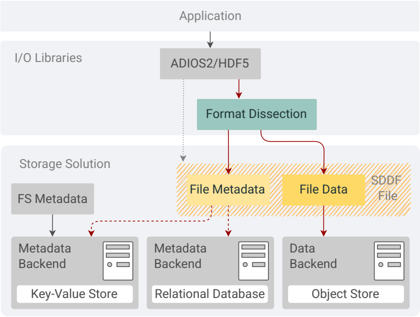


Format Dissection

Implementing Format Dissection

→ Map file metadata to storage backends

- Implementation in I/O library

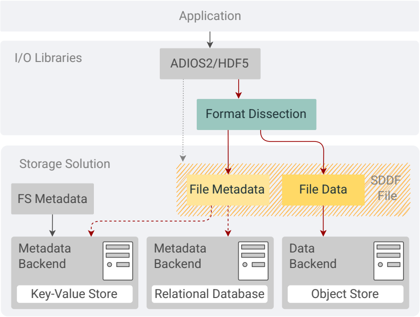


Format Dissection

Implementing Format Dissection

→ Map file metadata to storage backends

- Implementation in I/O library
- Transparent to application layer

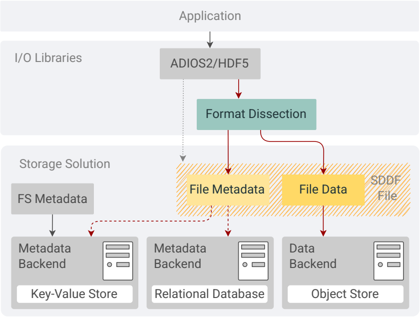


Format Dissection

Implementing Format Dissection

→ Map file metadata to storage backends

- Implementation in I/O library
- Transparent to application layer
- Using JULEA storage framework



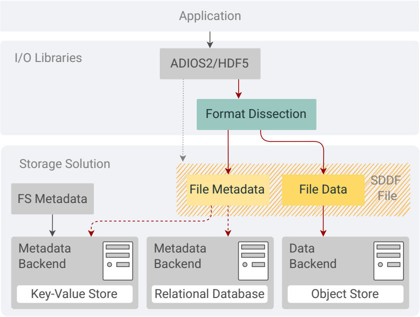
Format Dissection

Implementing Format Dissection

→ Map file metadata to storage backends

- Implementation in I/O library
- Transparent to application layer
- Using JULEA storage framework

- 1. prototype uses key-value store



Format Dissection

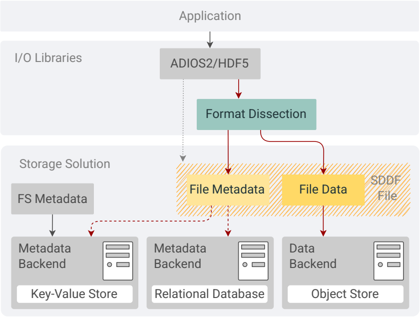
Implementing Format Dissection

→ Map file metadata to storage backends

- Implementation in I/O library
- Transparent to application layer
- Using JULEA storage framework

- 1. prototype uses key-value store

→ Demonstrate feasibility



Format Dissection

Refining Prototype

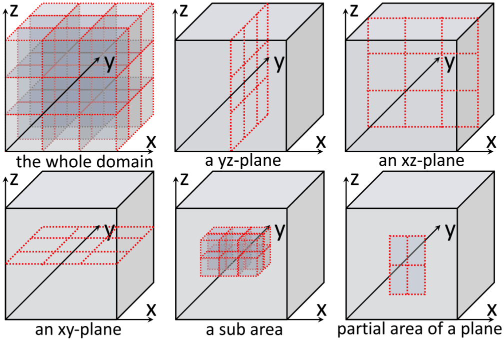
- How can metadata and data be queried efficiently?
- What custom metadata is interesting?

Access Patterns

- Various patterns in literature

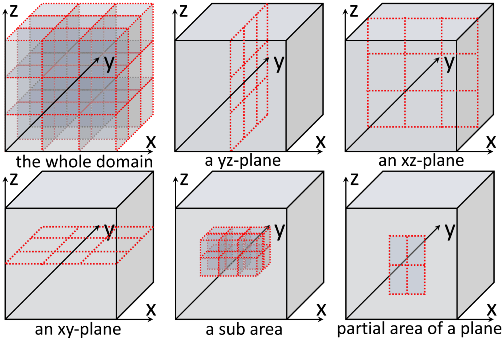
Access Patterns

- Various patterns in literature



Typical access patterns [Wan et al., 2022]

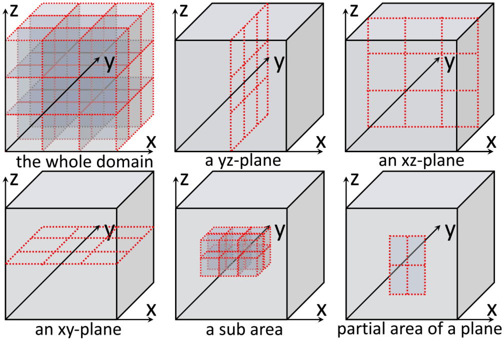
- Various patterns in literature
  → queries not clearly defined



Typical access patterns [Wan et al., 2022]

Access Patterns

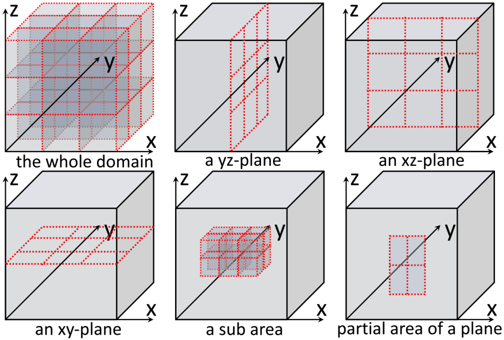
- Various patterns in literature
  → queries not clearly defined
- Data layouts can be complicated



Typical access patterns [Wan et al., 2022]

Access Patterns

- Various patterns in literature
  → queries not clearly defined
- Data layouts can be complicated
  → especially after dynamic load balancing

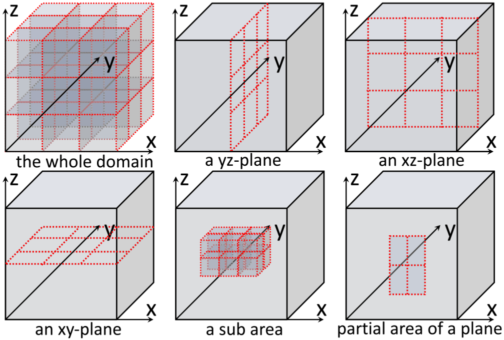


Typical access patterns [Wan et al., 2022]

Access Patterns

- Various patterns in literature
  → queries not clearly defined
- Data layouts can be complicated
  → especially after dynamic load balancing

→ General-purpose solution required



Typical access patterns [Wan et al., 2022]

Access Patterns

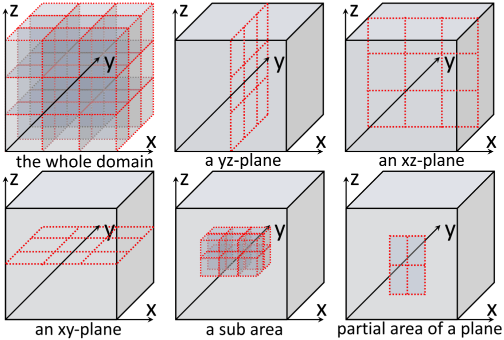
- Various patterns in literature

  → queries not clearly defined
- Data layouts can be complicated

  → especially after dynamic load balancing

→ General-purpose solution required

2. prototype uses a relational database



Typical access patterns [Wan et al., 2022]

User Survey

[a] Among others: DKRZ, MPI, DLR, DESY, DDN, SNL, LBNL, McGill

- Researchers from different institutes [a]

[a] Among others: DKRZ, MPI, DLR, DESY, DDN, SNL, LBNL, McGill

User Survey

- Researchers from different institutes [a]
- 22 completed, 38 in total

[a] Among others: DKRZ, MPI, DLR, DESY, DDN, SNL, LBNL, McGill

User Survey

- Researchers from different institutes [a]
- 22 completed, 38 in total
- 1. Format usage

[a] Among others: DKRZ, MPI, DLR, DESY, DDN, SNL, LBNL, McGill

User Survey

- Researchers from different institutes [a]
- 22 completed, 38 in total
- 1. Format usage

  → Informed database schema design

[a] Among others: DKRZ, MPI, DLR, DESY, DDN, SNL, LBNL, McGill

User Survey

- Researchers from different institutes [a]
- 22 completed, 38 in total
- 1. Format usage

  $\longrightarrow$ Informed database schema design
- 2. Post-processing

[a] Among others: DKRZ, MPI, DLR, DESY, DDN, SNL, LBNL, McGill

User Survey

- Researchers from different institutes [a]
- 22 completed, 38 in total
- 1. Format usage

  $\longrightarrow$ Informed database schema design
- 2. Post-processing

  $\longrightarrow$ Guided custom metadata choice

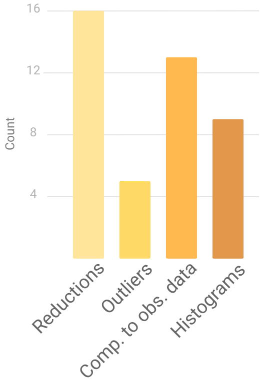
[a] Among others: DKRZ, MPI, DLR, DESY, DDN, SNL, LBNL, McGill

User Survey

- Researchers from different institutes [a]
- 22 completed, 38 in total
- 1. Format usage

  → Informed database schema design
- 2. Post-processing

  → Guided custom metadata choice



Typical post-processing operations

[a] Among others: DKRZ, MPI, DLR, DESY, DDN, SNL, LBNL, McGill

Data Analysis Interface (DAI)

Data Analysis Interface (DAI)

- Precompute common post-processing operations (default: mean and sum)

  $\longrightarrow$ *Reduce data access for analysis*

Data Analysis Interface (DAI)

- Precompute common post-processing operations (default: mean and sum)

  → *Reduce data access for analysis*
- Tagging functionality

  → *Mark interesting data parts*

Data Analysis Interface (DAI)

- Precompute common post-processing operations (default: mean and sum)

  → *Reduce data access for analysis*
- Tagging functionality

  → *Mark interesting data parts*
- Read

  → *Enable querying*

Data Analysis Interface (DAI)

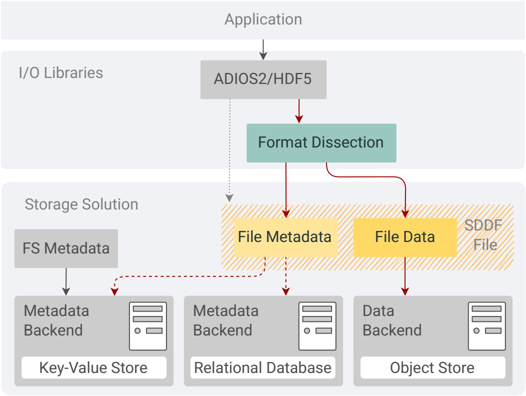
- Precompute common post-processing operations (default: mean and sum)

  → *Reduce data access for analysis*
- Tagging functionality

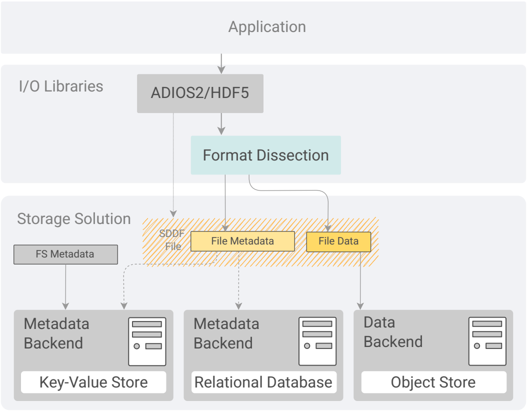
  → *Mark interesting data parts*
- Read

  → *Enable querying*

- Precompute common post-processing operations (default: mean and sum)
  - → *Reduce data access for analysis*
- Tagging functionality
  - → *Mark interesting data parts*
- Read
  - → *Enable querying*



- Precompute common post-processing operations (default: mean and sum)
  - $\longrightarrow$ *Reduce data access for analysis*
- Tagging functionality
  - $\longrightarrow$ *Mark interesting data parts*
- Read
  - $\longrightarrow$ *Enable querying*

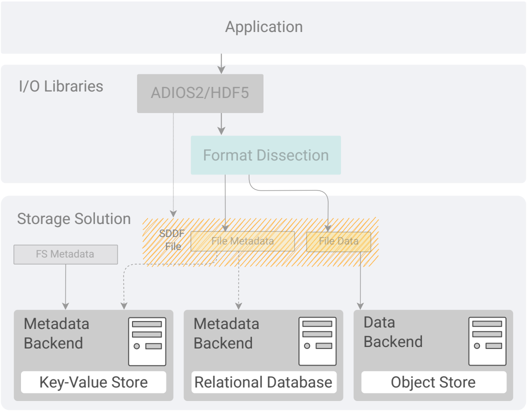


- Precompute common post-processing operations (default: mean and sum)

  → *Reduce data access for analysis*
- Tagging functionality

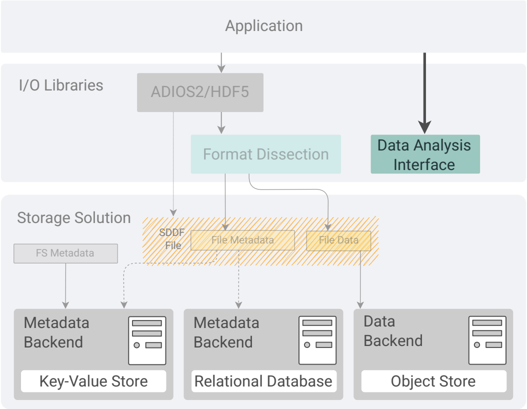
  → *Mark interesting data parts*
- Read

  → *Enable querying*



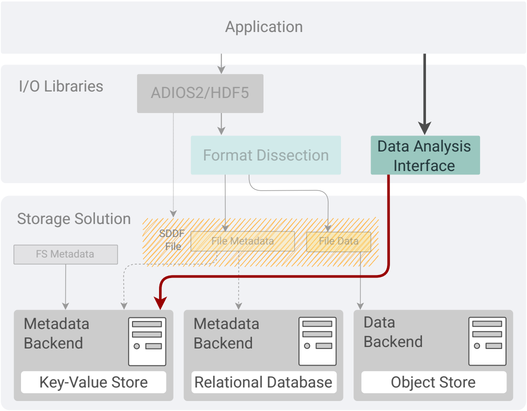
Data Analysis Interface (DAI)

- Precompute common post-processing operations (default: mean and sum)
  → *Reduce data access for analysis*
- Tagging functionality
  → *Mark interesting data parts*
- Read
  → *Enable querying*



Data Analysis Interface (DAI)

- Precompute common post-processing operations (default: mean and sum)
  → *Reduce data access for analysis*
- Tagging functionality
  → *Mark interesting data parts*
- Read
  → *Enable querying*



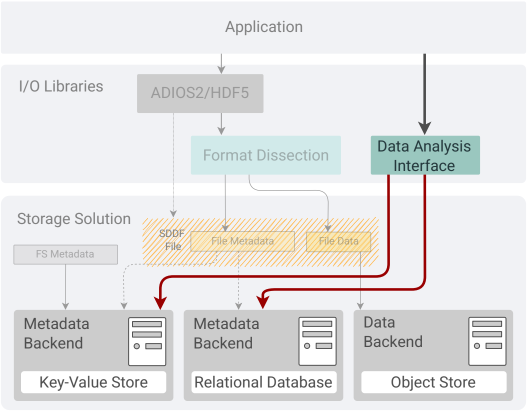
Data Analysis Interface (DAI)

- Precompute common post-processing operations (default: mean and sum)

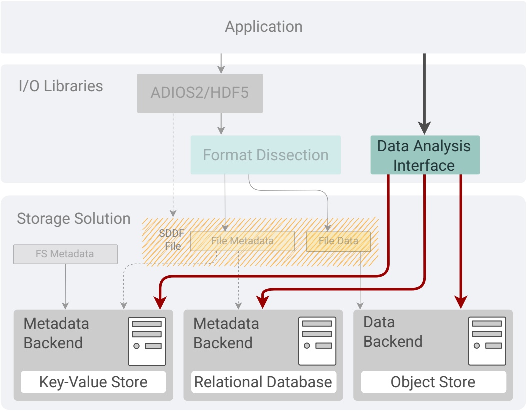
  → *Reduce data access for analysis*
- Tagging functionality

  → *Mark interesting data parts*
- Read

  → *Enable querying*



- Precompute common post-processing operations (default: mean and sum)
  → *Reduce data access for analysis*
- Tagging functionality
  → *Mark interesting data parts*
- Read
  → *Enable querying*

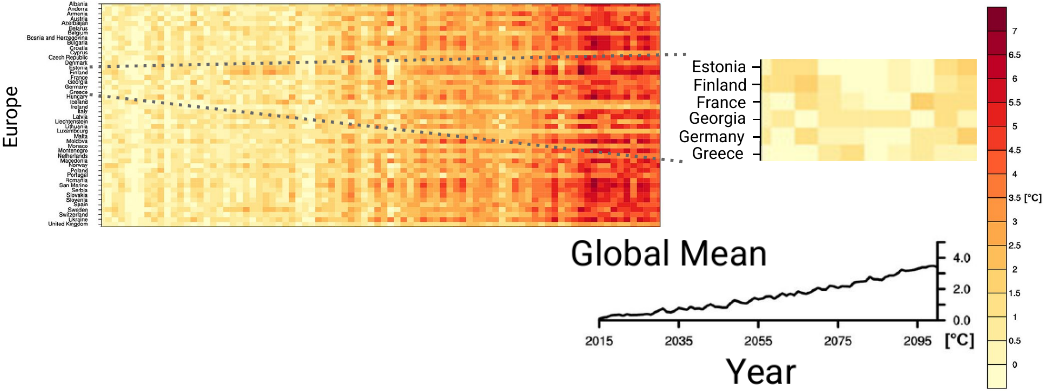


- Aggregations typical for climate research

Post-Processing Use-Case

- Aggregations typical for climate research

- Aggregations typical for climate research

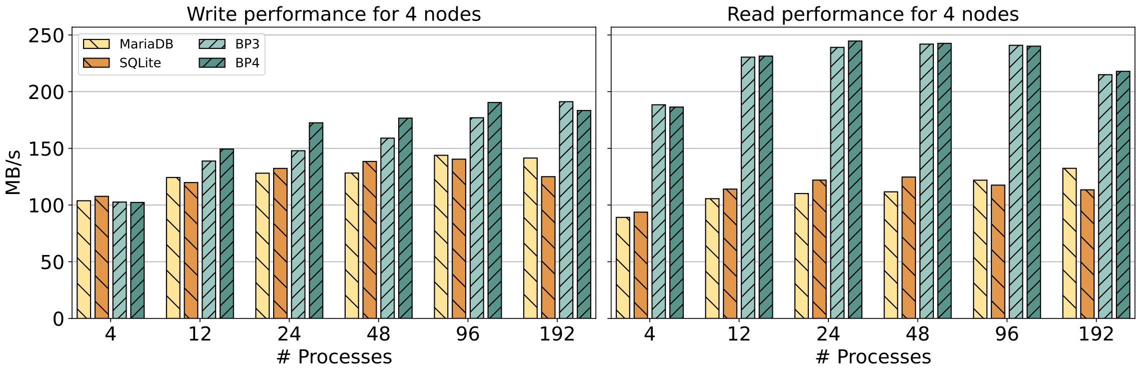


Outline

Motivation

Design

Evaluation



published at SYSTOR 2021

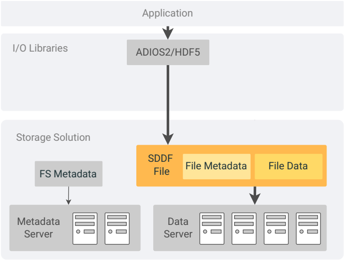
Evaluating the DAI

- Examine performance for different queries

Evaluating the DAI

- Examine performance for different queries
- Comparing different I/O paths

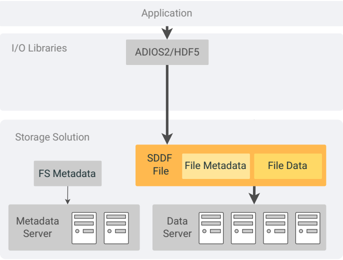
- Examine performance for different queries
- Comparing different I/O paths



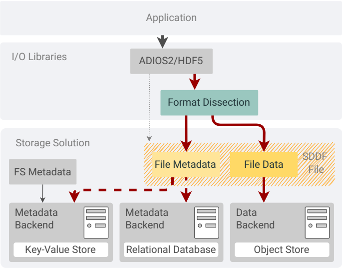
BP

Evaluating the DAI

- Examine performance for different queries
- Comparing different I/O paths

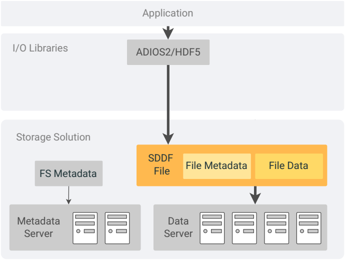


BP

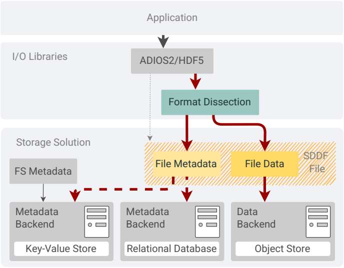


JULEA

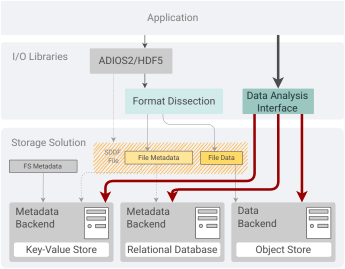
- Examine performance for different queries
- Comparing different I/O paths



BP



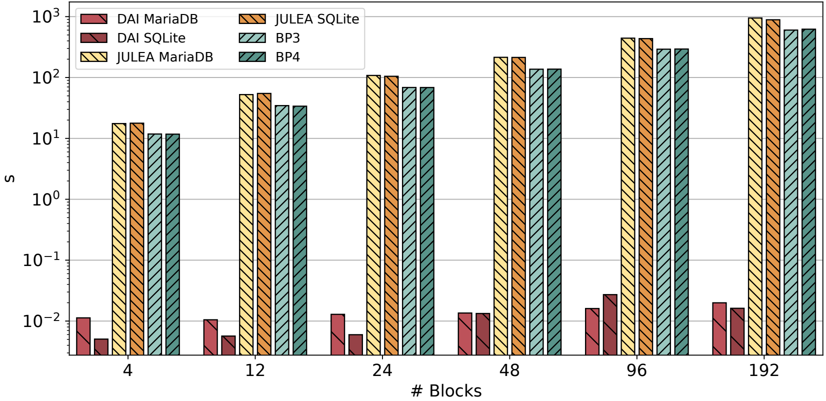
JULEA



DAI

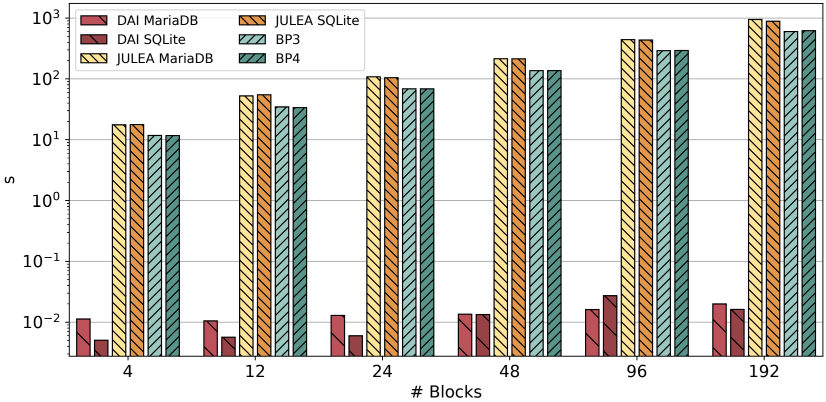
→ Retrieve largest difference between block mean value in step 1 and 5

Query Performance (Published in Duwe and Kuhn, SYSTOR'21)



→ Retrieve largest difference between block mean value in step 1 and 5

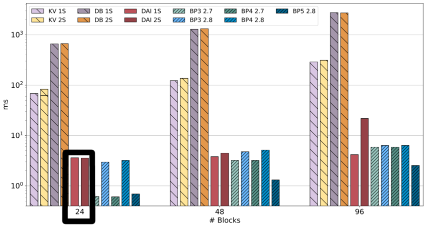
Query Performance (Published in Duwe and Kuhn, SYSTOR'21)



→ Retrieve largest difference between block mean value in step 1 and 5

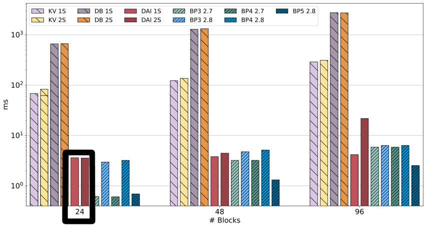
- For 192 blocks: 0.01 s (DAI) and 621/601 s (BP3/BP4)

No one size fits all

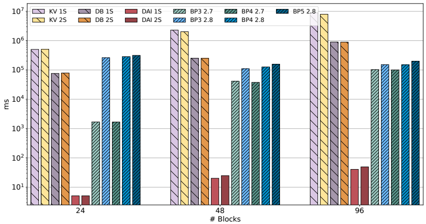


Query 1

No one size fits all

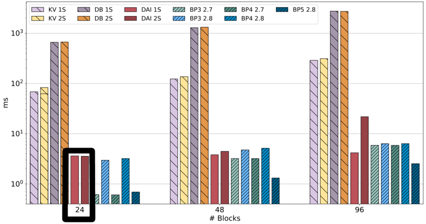


Query 1

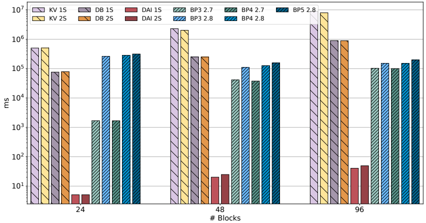


Query 2

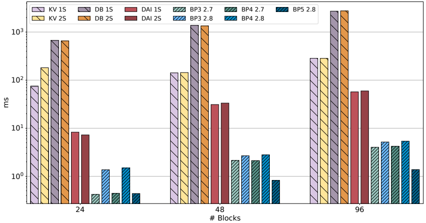
No one size fits all



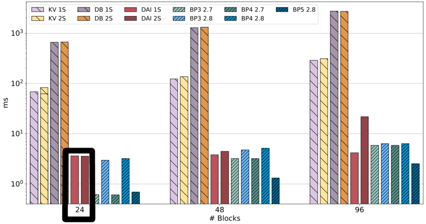
Query 1



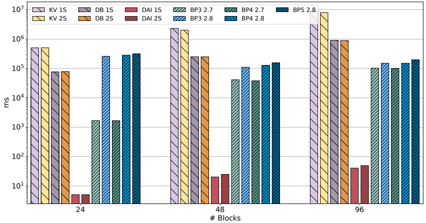
Query 2



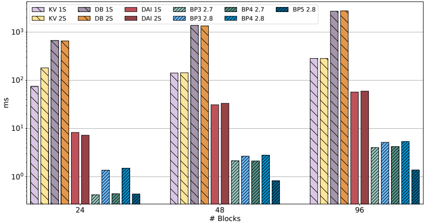
Query 3



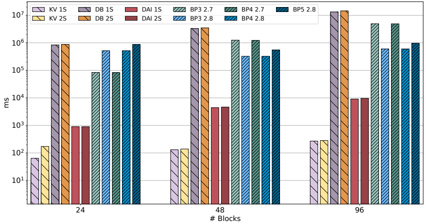
Query 1



Query 2



Query 3



Query 4

Generalising the Approach

Generalising the Approach

- Works for more than one format [Kuhn and Duwe, CSCI 2020]

Generalising the Approach

- Works for more than one format [Kuhn and Duwe, CSCI 2020]
- Applicable to different scientific fields

Generalising the Approach

- Works for more than one format [Kuhn and Duwe, CSCI 2020]
- Applicable to different scientific fields
- Problem: data models differ, e.g.

Generalising the Approach

- Works for more than one format [Kuhn and Duwe, CSCI 2020]
- Applicable to different scientific fields
- Problem: data models differ, e.g.
  - All data in a single 4D dataset

Generalising the Approach

- Works for more than one format [Kuhn and Duwe, CSCI 2020]
- Applicable to different scientific fields
- Problem: data models differ, e.g.
  - All data in a single 4D dataset
  - A separate 3D dataset for each step

Generalising the Approach

- Works for more than one format [Kuhn and Duwe, CSCI 2020]
- Applicable to different scientific fields
- Problem: data models differ, e.g.
  - All data in a single 4D dataset
  - A separate 3D dataset for each step
  - A separate file for each step

Generalising the Approach

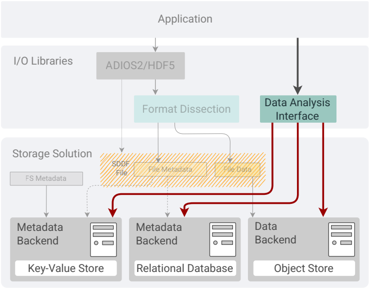
- Works for more than one format [Kuhn and Duwe, CSCI 2020]
- Applicable to different scientific fields
- Problem: data models differ, e.g.
  - All data in a single 4D dataset
  - A separate 3D dataset for each step
  - A separate file for each step

→ Either significant abstractions or restrictive modelling

Generalising the Approach

- Works for more than one format [Kuhn and Duwe, CSCI 2020]
- Applicable to different scientific fields
- Problem: data models differ, e.g.
  - All data in a single 4D dataset
  - A separate 3D dataset for each step
  - A separate file for each step

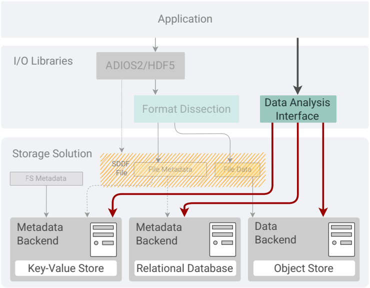
  → Either significant abstractions or restrictive modelling
- Implementation for every library → not easily scalable



Application
I/O Libraries
ADIOS2/HDF5
Format Dissection
Data Analysis Interface
Storage Solution
SDDF File
File Metadata
File Data
FS Metadata
Metadata Backend
Key-Value Store
Metadata Backend
Relational Database
Data Backend
Object Store

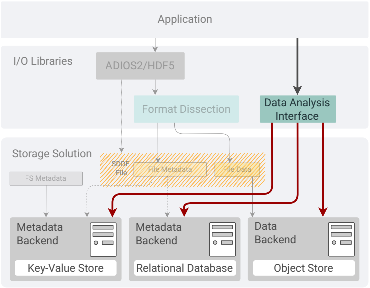
Conclusion

- Dissecting formats enables new management options



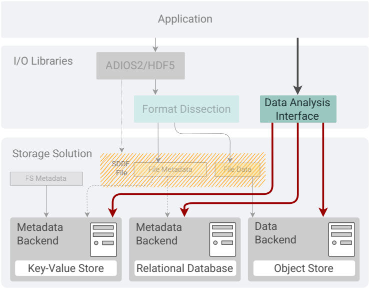
Conclusion

- Dissecting formats enables new management options
- No changes in simulation codes required



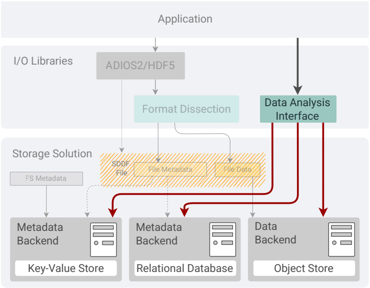
Conclusion

- Dissecting formats enables new management options
- No changes in simulation codes required
- Pre-computation of statistics can be very beneficial



Conclusion

- Dissecting formats enables new management options
- No changes in simulation codes required
- Pre-computation of statistics can be very beneficial
- DAI can achieve impressive performance (60,000 x)



Bibliography

[Duwe and Kuhn, 2021a] Duwe, K. and Kuhn, M. (2021a). **Dissecting Self-Describing Data Formats to Enable Advanced Querying of File Metadata.** In Proceedings of the 14th ACM International Conference on Systems and Storage, SYSTOR '21, New York, NY, USA. Association for Computing Machinery.

[Duwe and Kuhn, 2021b] Duwe, K. and Kuhn, M. (2021b). **Using Ceph's BlueStore as Object Storage in HPC Storage Framework.** In Proceedings of the Workshop on Challenges and Opportunities of Efficient and Performant Storage Systems, CHEOPS '21, New York, NY, USA. Association for Computing Machinery.

[Duwe et al., 2020] Duwe, K., Lüttgau, J., Mania, G., Squar, J., Fuchs, A., Kuhn, M., Betke, E., and Ludwig, T. (2020). **State of the Art and Future Trends in Data Reduction for High-Performance Computing.** Supercomput. Front. Innov., 7(1):4–36.

[Kuhn and Duwe, 2020] Kuhn, M. and Duwe, K. (2020). **Coupling Storage Systems and Self-Describing Data Formats for Global Metadata Management.** In 2020 International Conference on Computational Science and Computational Intelligence (CSCI), pages 1224–1230.

[Lüttgau et al., 2018] Lüttgau, J., Kuhn, M., Duwe, K., Alforov, Y., Betke, E., Kunkel, J. M., and Ludwig, T. (2018). **Survey of Storage Systems for High-Performance Computing.** Supercomput. Front. Innov., 5(1):31–58.

[Wan et al., 2022] Wan, L., Huebl, A., Gu, J., Poeschel, F., Gainaru, A., Wang, R., Chen, J., Liang, X., Ganyushin, D., Munson, T. S., Foster, I. T., Vay, J., Podhorszki, N., Wu, K., and Klasky, S. (2022). **Improving I/O performance for exascale applications through online data layout reorganization.** IEEE Trans. Parallel Distributed Syst., 33(4):878–890.