DASI: data access and storage interface

An interface for semantic data management

Jenny Wong, Metin Cakircali, Olivier Iffrig, Simon Smart, James Hawkes and Tiago Quintino

Forecast Department, ECMWF, Reading, United Kingdom

Abstract

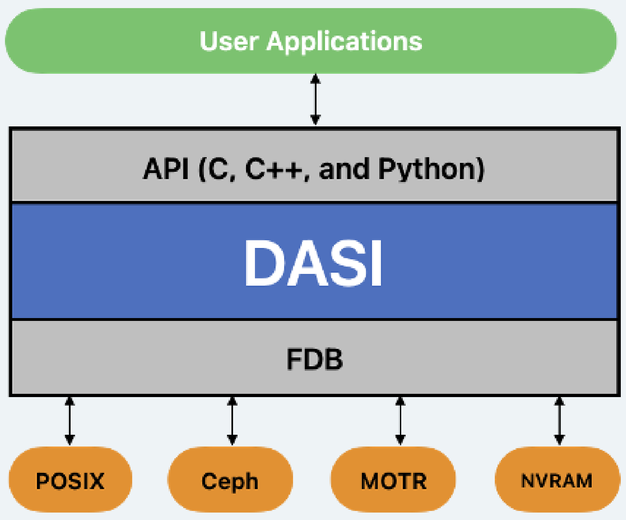
With supercomputers reaching exascale, new approaches to data management and storage aimed at eliminating unnecessary data movement are essential for I/O intensive workflows. In the IO-SEA project, which aims to implement solutions for scaling applications to exascale HPC systems, ECMWF have developed DASI, a Data Access and Storage Interface. The interface enables users to manage data using domain specific and scientifically meaningful metadata keys, and separates data management from the underlying backend storage technology through abstraction.

Concept

Often data is either labelled with a unique identifier, which provides no description of the contents, or is addressed in a storage implementation specific manner:

- ✕ `bucket/jdo92md83nnf302k`
- ✕ `/scratch/$USER/weather/20230101/experiment-42/rain`

This places a burden on the user in managing the data, which can be prone to error, and is not easily adaptable to different storage solutions. DASI provides an interface for managing data through a semantic description of the data, which is domain specific and scientifically meaningful.



Design

The design of DASI is based on the FDB software in operational use at ECMWF, but is domain agnostic, allowing it to be configured for use in any scientific domain. It is configured through collection schemas, which specify the semantic keys as well as their hierarchy. Each collection schema is a tree with three levels, where each level can contain any number of keys, and data is addressed by specifying all the keys along a path from the root to a leaf.

```
model, number
  date, sector
    variable
  epoch, location
    level
```

Collection Schema

```
model: weather
number: 42
date: 20230101
sector: west
variable: rain
```

```
model: weather
number: 42
epoch: 200
location: uk
level: 250
```

Valid Addresses

The tiered structure of the schema allows the user to specify how the data should be organised. DASI abstracts the underlying storage technology to enable the user to use a single interface for different backends, and allows the user to relinquish the responsibility of arranging the data to DASI.

Data can be queried and retrieved by specifying a subset of the keys, and a list of possible values for each key.

```
model:  weather
experiment:  [41, 42]
date:  [20230101, 20230101]
variable:  rain
```

Example Query

This format enables the user to retrieve multiple entries in a single query. Users are also able to set policies or hints regarding the data lifetime and usage frequency through DASI. These are passed directly to the storage backend which, if supported, allows users to make use of hierarchical storage management.

Usage

The API is available in C++, C and Python, and is also exposed through a command line interface. The main burden on the user is devising a suitable schema for their use-case, in particular, the separation of the key words into three levels of hierarchy.

```
# Collection 1
[model, number,    # Level 1:  specifies top level directory
  [date, sector,   # Level 2:  specifies filename
    [variable],    # Level 3:  indexes entries in file
  ],
  [epoch, location,
    [level]
  ]
]
# Collection 2
[institute, location,
  [date, method?, # "?" used for optional key specification
    [variable]
  ]
]
```

Once configured, the interface allows users to perform the following actions on their data:

archive list retrieve

As DASI is built on top of the FDB object store, it is able to make use of the different backend storage technologies the FDB has been adapted to use. In the simplest case of a POSIX backend, the DASI objects are stored as files, but other supported backends include Ceph object store, Cotx-Motr and NVRAM.

DASI Github: https://github.com/ecmwf-projects/dasi
DASI Documentation: https://dasi.readthedocs.io
FDB Github: https://github.com/ecmwf/fdb
IO-SEA Project: https://iosea-project.eu

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955811. The JU receives support from the European Union's Horizon 2020 research and innovation programme and France, the Czech Republic, Germany, Ireland, Sweden, and the United Kingdom.