Improving the I/O performance gap with MultIO –
Recent developments and performance studies

Stefanie Reuter[1*], Domokos Sarmany[1], Mirco Valentini[1], Razvan Aguridan[2,1], Philipp Geier[1], Tiago Quintino[1], Simon Smart[1]

(1) ECMWF; (2) Barcelona Supercomputing Center (*) Stefanie.reuter@ecmwf.int

1. Background

Current projections of data volumes produced in numerical weather prediction and climate simulations expect an increase to over a petabyte per day in the next few years. An efficient and customizable way to increase throughput to and from storage is needed.

2. MultIO as part of the Digital Twin Engine

The Digital Twin Engine (Figure 1) is a modular set of tools developed at ECMWF that builds the framework around a Digital Twin forecast providing data and control-flow paths. MultIO [1,2] is a set of C++ libraries that sit close to the model and receive messages with raw model data. MultIO offers two distinct functionalities:

- **I/O-server** functionality to create aggregated horizontal fields from distributed parallel models, decoupling compute from I/O.
- **Post-processing** pipelines to calculate derived products, such as temporal pointwise statistics, interpolation between different grids, data encoding and data output.

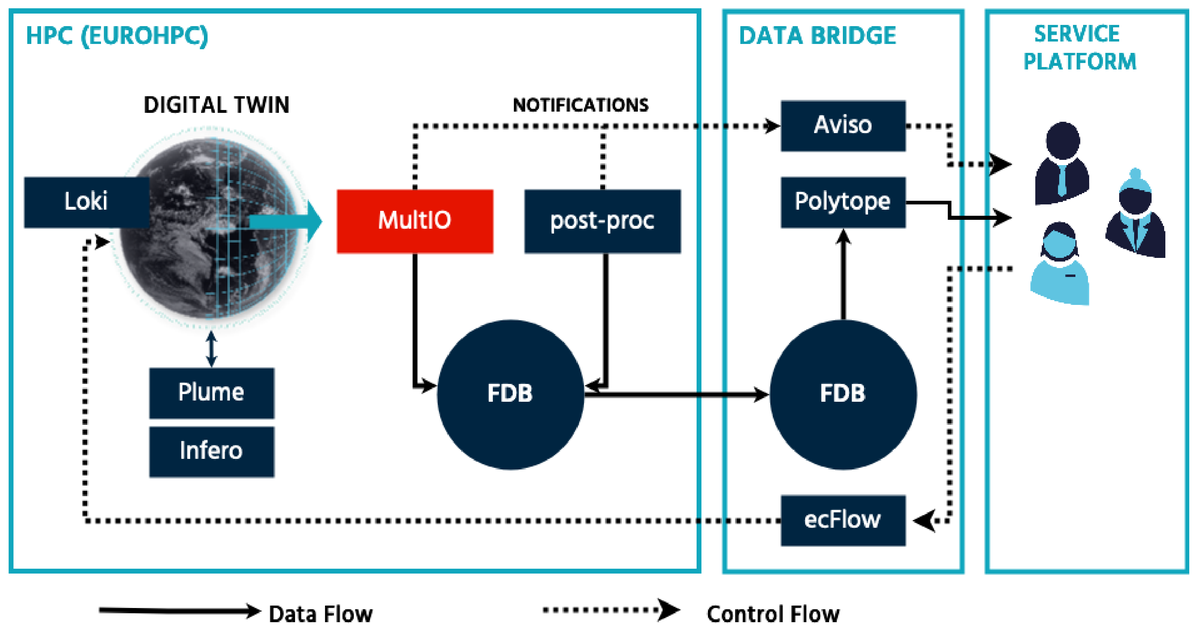

Figure 1: ECMWF's Digital Twin Engine

3. Message-driven data streaming

MultIO offers a modular approach to generate individual processing pipelines, consisting of a series of actions, carried out based on message metadata routing and action configurations. Users of the library can either create their own actions or choose from a set of pre-defined actions. Figure 2 shows an exemplary pipeline. On the client side a plan is selected, based on metadata information, to either perform temporal statistics or directly transport the message to the MultIO server. The I/O-server can be configured in a similar way with pass-through **on-the-fly actions**, such as interpolate, statistic, encoding and finalizing actions. The latter can be a sink action to save the data to storage, in this case to FDB [4], a domain-specific object store for meteorological data.

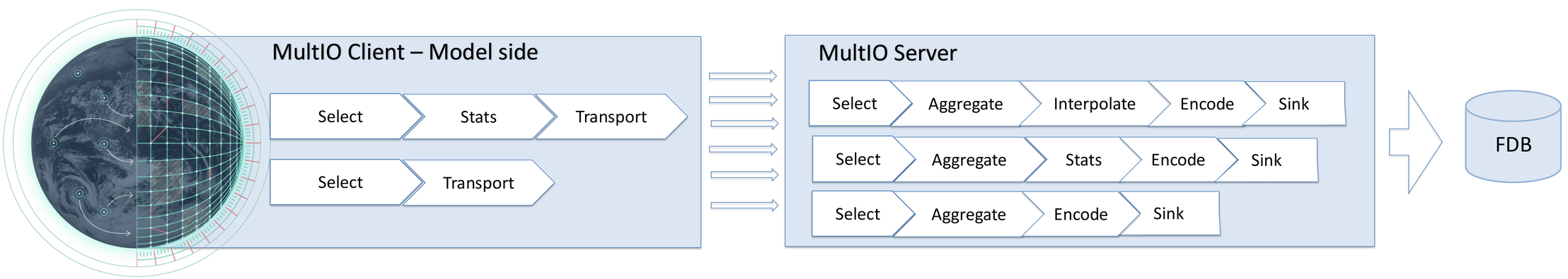

Figure 2: An example MultIO plan configuration with a model side MultIO client and a MultIO server where post-processing products are generated and passed on to storage.

4. Recent development and performance studies

Interpolation and Regridding

The interpolation library **MIR** [3,1] can be called directly on Model data whilst still in memory. This enables different output format transformations to the ones use during the computation, such as the HEALpix grid. To further improve the initialization phase, **MIR Caches** were introduced that allow interpolation weights/matrices to be reused between runs.

Recent optimizations

- Implementation of new metadata handling
- Sharing of metadata and payload among pipelines where possible
- Use of pre-hashed keys

To quantify recent optimizations, the I/O overhead in NEMOv4 was measured by timing all I/O routines with calls to MultIO. The following experiment setup was used:

- Compute a single trajectory with NEMOv4 at eORCA12 resolution
- On 200 compute nodes (3200 MPI tasks) and 128 I/O-server tasks
- Resulting in approximately 29,000 GRIB messages

Figure 3 shows the accumulated I/O overhead, visualized in orange, indicating a **performance gain of almost 70 %** compared to MultIO 2.1.9.

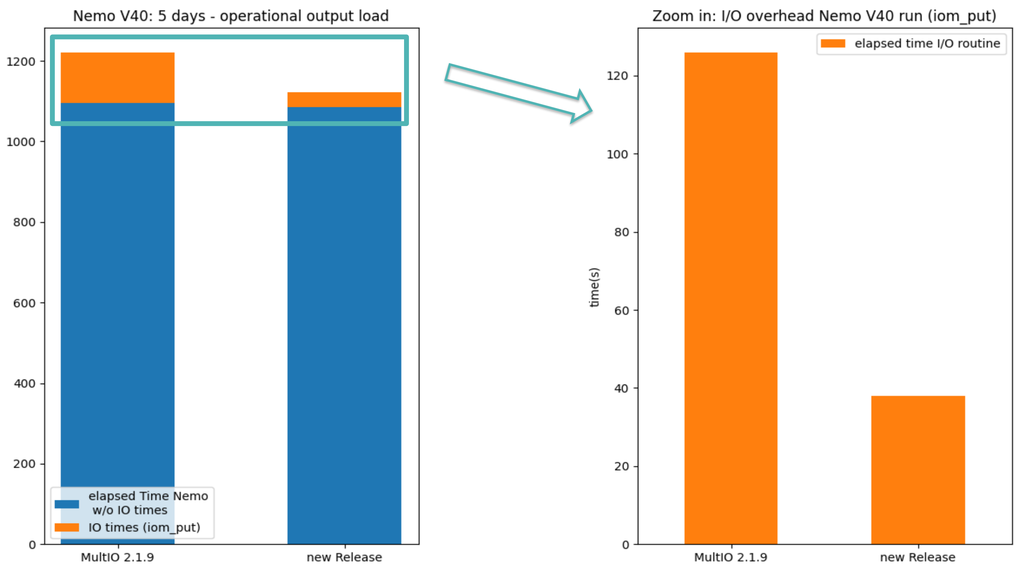

Figure 3 Visualizing the I/O overhead of a 5-day eORCA12 NEMOv4 ocean model run

Performance study: Ocean re-analysis to compare MultIO with XIOS25

Re-analyses are an important step to create a complete picture of the past, that combine observations with short-range weather modelling. MultIO provides the ability to encode ocean data to GRIB2 format, the global standard for meteorological data, as well as to compute temporal statistics. A performance study has been conducted to compare MultIO to XIOS 2.5 (Figure 4) in a best effort experiment with the following setup:

- Compute a single trajectory with different output volume configurations with NEMOv4
- Accumulated output of up to 1.5 TiB per run
- Computation and output at eORCA025 (quarter degree) resolution

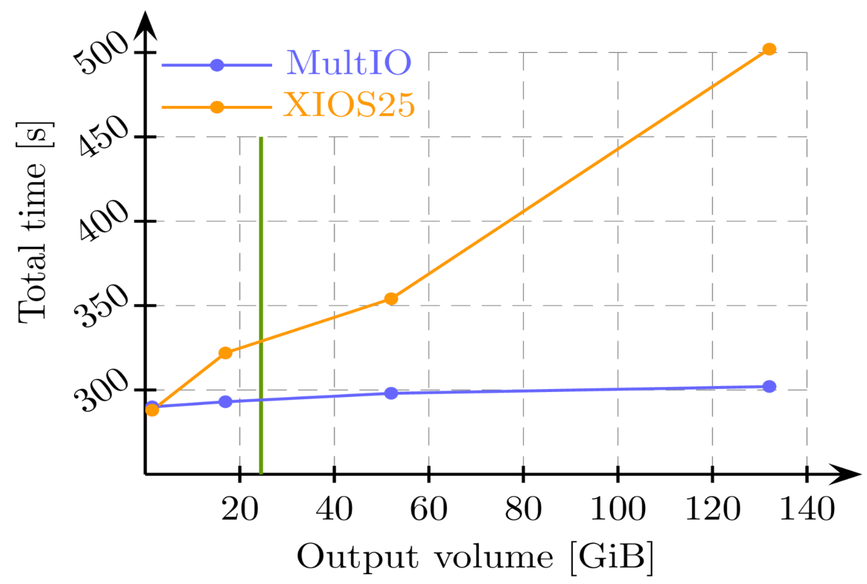

Figure 4: Comparison of MultIO 2.0 with XIOS 2.5 for different throughput volumes of one five-day one-ensemble-member assimilation loop. [1]

3. References

[1] Domokos Sarmany, Mirco Valentini, Pedro Maciel, Philipp Geier, Simon Smart, Razvan Aguridan, James Hawkes, and Tiago Quintino. 2024. MultIO: A Framework for Message-Driven Data Routing For Weather and Climate Simulations. In Proceedings of the Platform for Advanced Scientific Computing Conference (PASC '24). Association for Computing Machinery, New York, NY, USA, Article 24, 1–12. https://doi.org/10.1145/3659914.3659938

[2] https://github.com/ecmwf/multio

[3] https://github.com/ecmwf/mir

[4] Simon D. Smart, Tiago Quintino, and Baudouin Raoult. 2019. A High-Performance Distributed Object-Store for Exascale Numerical Weather Prediction and Climate. In Proceedings of the Platform for Advanced Scientific Computing Conference (PASC '19). Association for Computing Machinery, New York, NY, USA, Article 16, 1–11. https://doi.org/10.1145/3324989.3325726

Funded by the European Union Destination Earth implemented by ECMWF esa EUMETSAT