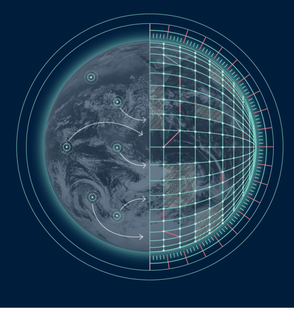
MultIO: An Open-Source Framework for Message-Driven Data Routing for Earth-System Models

Domokos Sármány, Mirco Valentini, Razvan Aguridan, Philipp Geier, James Hawkes, Simon Smart, Tiago Quintino

(1) ECMWF; (*) domokos.sarmany@ecmwf.int



1. MultIO pipelines

MultIO is a set of C++ libraries with both Fortran and C interfaces for **data routing** (data streaming) from distributed meteorological and **earth-system models**. It supports two related, but distinct functionalities.

- **Post-processing** to calculate derived meteorological products, such as temporal pointwise statistics, interpolation onto different grids, encoding data into output formats, and output of data to storage systems or other consumers.
- **I/O-server** functionality to create aggregated horizontal fields from distributed parallel models.

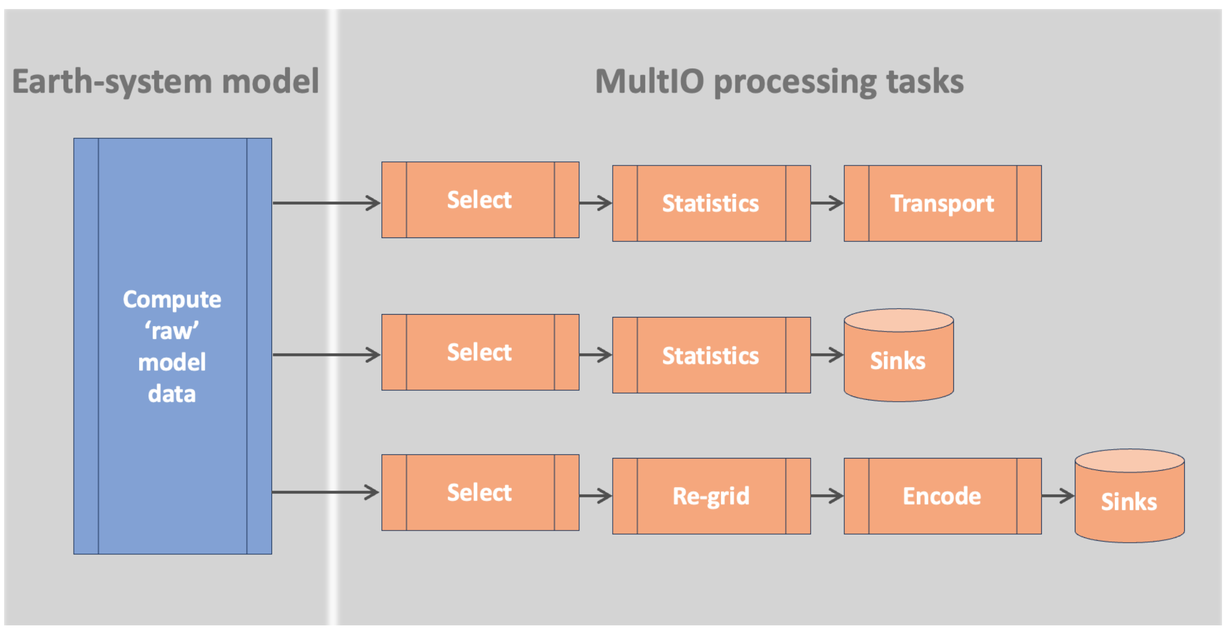


Illustration of processing pipelines for model data. The model interfaces with MultIO to pass every message to every pipeline. Each pipeline starts with a filtering 'Select' action and terminates with a 'Sink' or 'Transport', with an arbitrary number of post-processing actions in between.

2. Message-driven data routing (streaming)

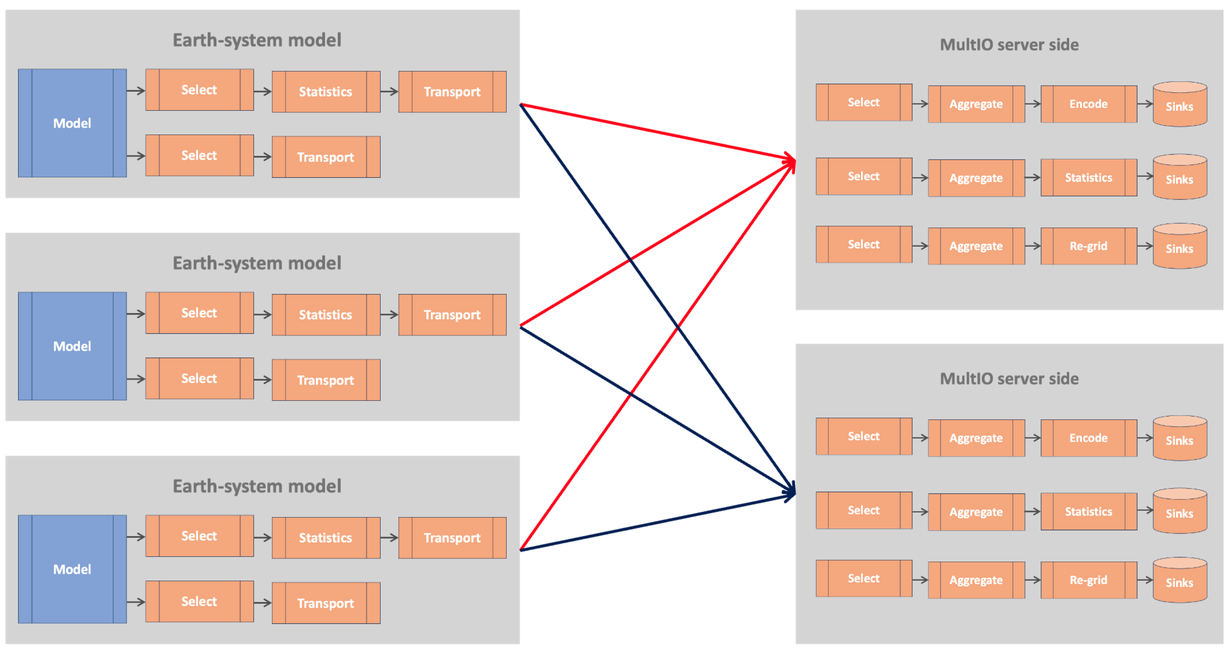
MultIO routes messages through **user-programmable** pipelines of actions. **Routing decisions** are based on the **message metadata** and **action configuration**. An action can be one of the following.

- **Select** to filter based on metadata key
- **Transport** to transfer data across the network
- **Aggregation** to create global fields (I/O-server)
- **Statistics** to compute temporal operations
  - Hourly, daily, monthly units for periods
  - Instant, average, min, max, accumulate
- **Metadata-mapping** between data models
- **Interpolate** to supported target grids
- **Cropping**
- **Encode** from raw data to given format (GRIB)
- **Sink** to object store, filesystem

3. MultIO as an I/O-server

MultIO is also a **generic I/O-server** for distributed parallel models.

- Implemented as a **client-server architecture**.
- Two sets of pipelines:
  - **Model (client)** side, acting on **partial fields**, terminating in a transport action sending (partial) fields to the servers,
  - **Server** side with a dedicate action to **aggregate** partial fields, and further actions on **global fields**.
- **Asynchronous** communication between model (client) and server.
- Support communicating **partial-to-global mappings** for aggregation.
- Support communicating **mask information** for more efficient data storage.



4. Usage in the NEMOv4

For use in **Destination Earth**'s Climate **Digital Twin** (DT).

- Computations of **temporal means** on the model (client) side.
- **Aggregate** partial fields based on **ORCA grids** mapping information.
- **Mask** land points.
- **Encode** global fields in **GRIB2**.

```yaml
client:
  plans:
    - name: ocean-domain-info
      actions:
        - type: select
          match:
            - category: [ocean-domain-map, ocean-mask]

        - type: transport
          target: server

    - name: stream-1
      actions:
        - type: select
          match:
            - name: [ sss, sst, ice_cover ]

        - type: metadata-mapping
          mapping: '{~}/metadata-mapping/nemo-to-grib.yaml'
          enforce-match: true

        - type: statistics
          output-frequency: 3h
          operations:
            - average

        - type: transport
          target: server
```

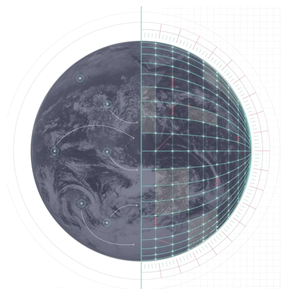
```yaml
server:
  transport: mpi
  plans:

    - name: ocean-fields
      actions:
        - type: select
          match:
            - category: [ocean-2d, ocean-3d]

        - type: aggregate

        - type: mask
          apply-bitmap: true
          offset-fields: [sst, toce, icettop]
          offset-value: 273.15

        - type: encode
          format: grib
          template: "{~}/unstr_avg_fc.tmpl"
          grid-type: eORCA12
          run:
            expver: dsfc # bilg
            class: rd   # rd
            stream: oper
            type: fc

        - type: sink
          sinks:

            - type: fdb

            - type: file
              append: true
              per-server: true
              path: ocean-output-field.grib
```

5. Usage in coupled atmosphere-ocean models

For DE's **Climate DT** and the EU's **nextGEMS** projects.

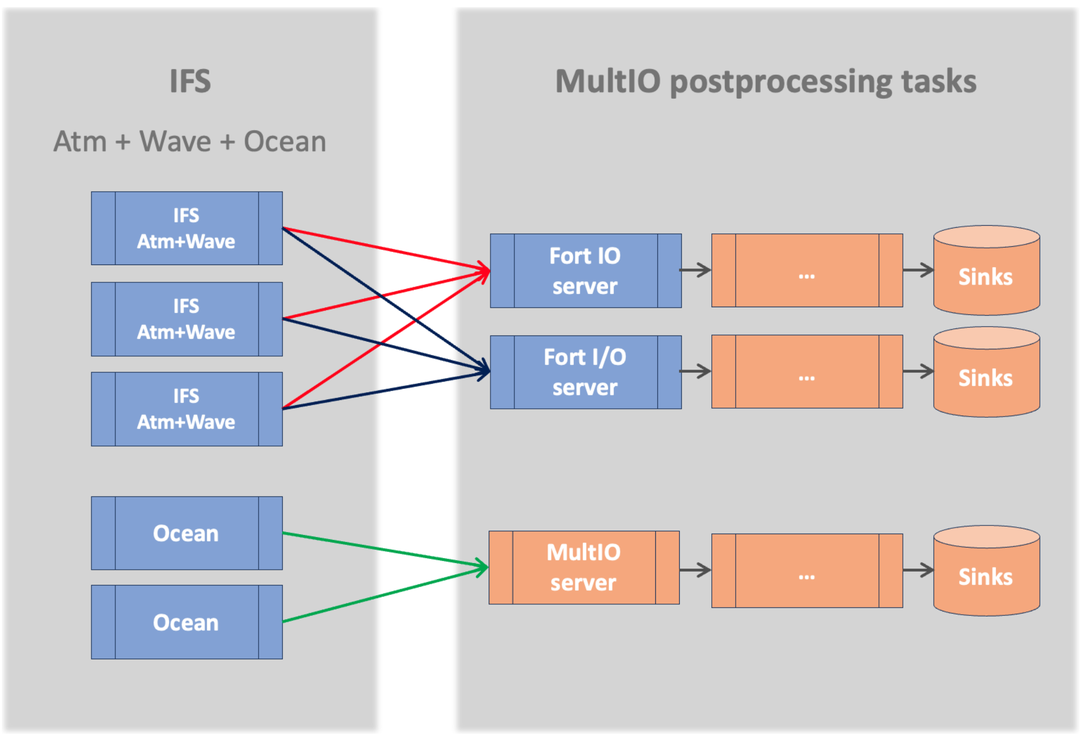
NextGEMS multi-year runs

- **Coupled** to NEMOv3 with no output
- **Post-processing** pipeline for IFS
- **Statistics** (monthly means)
- **Interpolation** (re-gridding)



High-resolution DestinE climate runs

- **Two** coupled **ocean-atmosphere** models:
  - IFS/NEMOv4
  - IFS/FESOMv2
- **I/O-server** for both NEMOv4 and FESOMv2
- **Uniform** output on HEALPix grids (both ring and nested ordering)
- **Encode** post-processed fields in **GRIB2.**
- **Integrate** output configuration



6. Summary of key benefits

MultIO benefits earth-system models by:

- **Asynchronous I/O**, not blocking during data output
- **Moving computation closer to data**, especially when data is only needed computation
- Flexible design **for chaining post-processing acti**
- **Message-driven** data routing (data streaming)
- **Open source**: https://github.com/ecmwf/multio.

Funded by the European Union

Destination Earth

implemented by ECMWF esa EUMETSAT