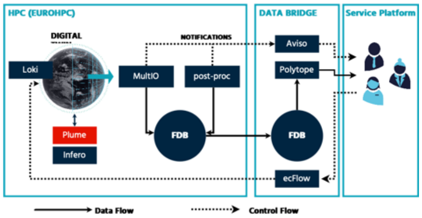
Plume Plugin System: An Overview of the Recent Developments

Antonino Bonanni[1*], Domokos Sarmany[1], James Hawkes[1], Tiago Quintino[1]
(1) ECMWF; (*) antonino.bonanni@ecmwf.int

1. Plume: Plugin Mechanism

Under the Destination Earth initiative, the European Centre for Medium-Range Weather Forecasts (ECMWF) is developing a plugin system for Earth-system models called Plume. Plume provides a model with additional data processing functionalities through plugins.



Plugins are separate software libraries that are loaded at runtime and through the Plume interface can access model data in memory (when data is also potentially distributed across several computing nodes in the HPC system). Plugins can then implement "on-the-fly" data processing functionalities without the need for data only required by those calculations to be written to disk.

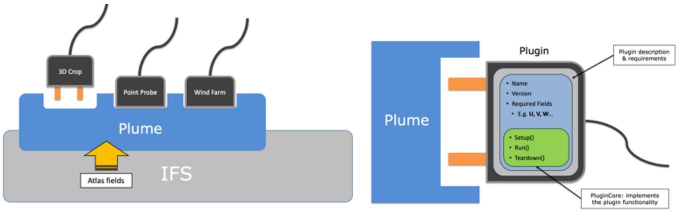


Diagram of plume library interfaced with ECMWF Integrated Forecasting System (IFS). Numerical fields are passed on to plugins through the Atlas library. On the right, the schematic structure of a plume plugin.

This design guarantees the necessary system scalability, especially considering the high-resolution adopted in the DestinE Digital Twins. Additionally, a system designed for plugins guarantees modularity and facilitates collaborative development across organizations. In fact, once the interface to model data is defined in Plume, a plugin can effectively be developed independently from the model in which will be used.

Plume is part of the DestinE Digital Twin Engine and is Open Source via the ECMWF GitHub space (github.com/ecmwf/plume).

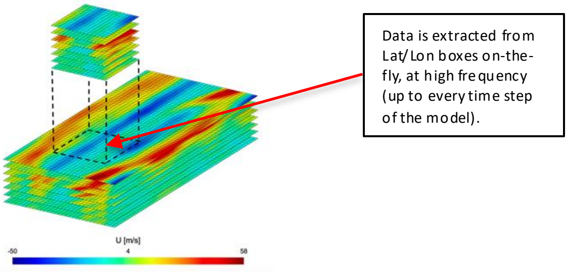
2. Plugin Development

Plugin development is key for the plugin mechanism. Accessing the model data is achieved through the Plume interface. Upon requesting specific parameters, a plugin can make use of the model data at runtime on-the-fly. The actual Plume interface is accessible through C, C++ and Fortran programming languages.

The structure of a plugin is simple by design (figure above). It consists of a "setup" function that is called by the model during its setup phase (before time stepping), a "run" function called at each step of the model and a "teardown" function called at the end of the simulation. The plugin can operate on model data in each phase and perform the desired data-processing task.

3. Data Extraction Plugin

One of the recent developments is a plugin that can be configured to extract data from a pre-defined region and write the cutout data directly in user-friendly CovJSON format. This plugin allows high-frequency data extraction from local areas (potentially up to every time integration step of the model). This plugin has been tested in a dedicated run of the DestinE ExtremesDT.

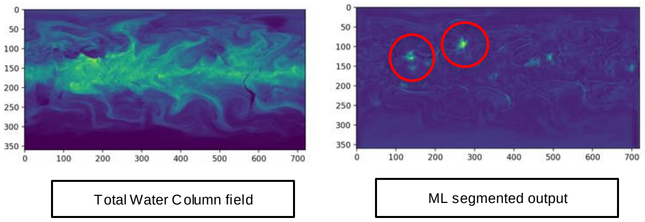


Data is extracted from Lat/Lon boxes on-the-fly, at high frequency (up to every time step of the model).

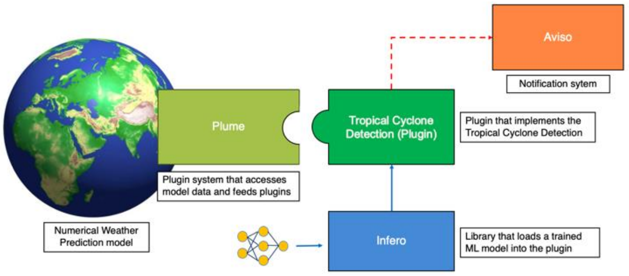
4. Machine Learning Tropical Cyclone Detection Plugin

Another plugin recently developed performs Tropical Cyclone detection by running a Machine Learning model on-the-fly. The plugin loads a pre-trained ML model during the setup phase and then runs it for inference at each time step of the simulation. The handling of the Machine Learning model inside the plugin is delegated to a separate library called Infero. An example is shown in the picture below, where the plugin was tested on an IFS (RAPS) simulation. In this example, a notification is also sent to an Aviso server when a Tropical Cyclone is detected.

The Machine Learning inference library Infero (github.com/ecmwf/infero) and the notification system Aviso (github.com/ecmwf/aviso) are also part of the DestinE Digital Twin Engine.



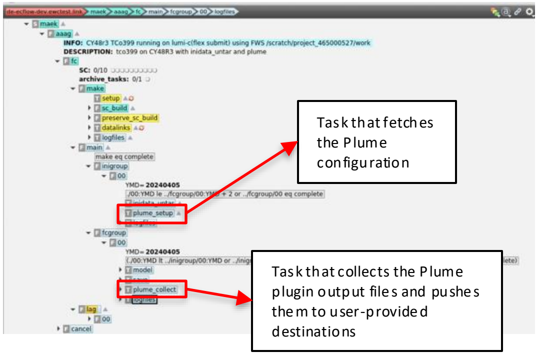
Tropical Cyclone detection algorithm implemented in a Plume plugin through the Digital Twin Engine library Infero.


Aviso
Notification sytem
Plume
Plugin system that accesses model data and feeds plugins
Tropical Cyclone Detection (Plugin)
Plugin that implements the Tropical Cyclone Detection
Numerical Weather Prediction model
Infero
Library that loads a trained ML model into the plugin

Overview of a prototype system showcasing a Tropical Cyclone detection plugin that also sends notification to Aviso.

5. Integration of Plume in the ExtremesDT Suite

At the end of DestinE phase-1, Plume was integrated into an ad-hoc ExtremesDT suite as part of a capability demonstration run. Users could configure specific areas from which to retrieve wind data (in CovJSON format) and receive the data directly into provided remote destinations. This prototype mechanism was tested both on the ECMWF HPC system and on the EuroHPC system LUMI.



```
maek
  aaag
    INFO: CY48r3 TCo399 running on lumi-c(flex submit) using FWS /scratch/project_465000527/work
    DESCRIPTION: tco399 on CY48R3 with inidata_untar and plume
    fc
      SC: 0/10
      archive_tasks: 0/1
      make
        setup
        sc_build
        preserve_sc_build
        datalinks
        logfiles
      main
        make eq complete
        inigroup
          00
            YMD= 20240405
            ./00:YMD le ../fcgroup/00:YMD + 2 or ../fcgroup/00 eq complete
            inidata_untar
            plume_setup
            logfiles
        fcgroup
          00
            YMD= 20240405
            (../00:YMD lt ../inigroup/00:YMD or ../inig                    lete)
            model
            save
            plume_collect
            logfiles
      lag
        00
    cancel
```

Task that fetches the Plume configuration

Task that collects the Plume plugin output files and pushes them to user-provided destinations

Plume integrated in a capability demonstration run of the ExtremesDT suite. Tests have been performed in the ECMWF HPC system and in the EuroHPC system LUMI.

6. Conclusions

In the past year, Plume has been developed further and additional plugins have been added to the existing collection of Plume plugins (some of which are already been made open source). In addition, the Plume mechanism is currently in the process of being integrated into DestinE ExtremesDT suites.

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