Enhancing ARGO floats data re-usability

Gianpaolo Coro, ISTI-CNR (Italy), *gianpaolo.coro@isti.cnr.it*
Paolo Scarponi, ISTI-CNR (Italy), scarponi@isti.cnr.it
Pasquale Pagano, ISTI-CNR (Italy), pasquale.pagano@isti.cnr.it

Many research communities in a great variety of fields are interested in accessing collections of reliable environmental data. These data are typically used in environmental monitoring systems, data processing workflows, ecological models, societal and economical analyses, etc. Research communities need to carry out their studies in a fast and efficient manner and thus require data to be well structured, well described, and possibly represented in standard formats that allow direct access and usage. In this context, reducing data preparation and pre-processing time is crucial.

ARGO data have been long-used by marine science communities in global oceans observing systems. These data are collected using a large network of floats, monitored by the ARGO Information Center (AIC) and are sent to Global Data Assembly Centers (GDACs). The datasets are available for download on the official ARGO website (www.argo.ucsd.edu), as Network Common Data Format (NetCDF) *Point-feature* files and CSV files through FTP sites and online tools. However, these formats present many challenges from a technical point of view, especially in terms of re-usability. Every dataset has dimension ranging from 5MB to 3GB and contains measurements in time of different physical parameters recorded at different locations. Every file corresponds to one month and the overall repository time-span ranges from January 1999 to today. An overall CSV repository is available (ftp://ftp.ifremer.fr/ifremer/coriolis/co0547-bigdata-archive/) where a JSON file stores metadata about the parameters, e.g. the unit of measure, the full name, the reliability of the measurement, etc. Although accessing this unique endpoint is convenient, every dataset is not a standalone object and requires continuously parsing the JSON file to be fully understood. Further, managing a 3GB CSV file can be memory demanding, especially for processes that need to combine this dataset with other data.

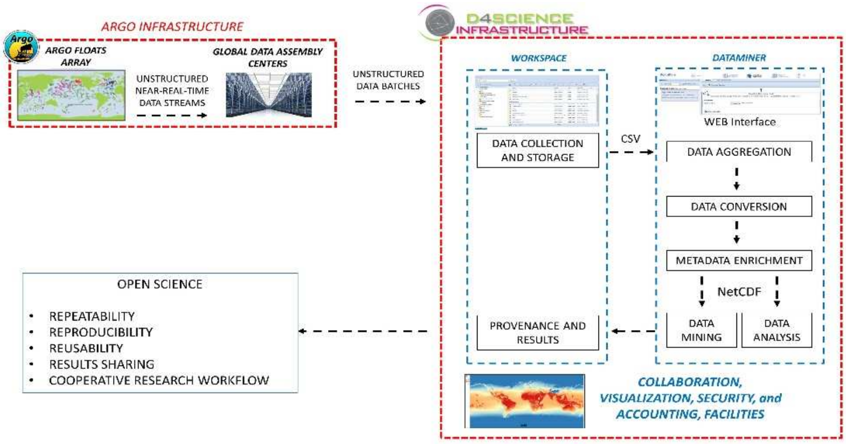

Fig. 1: Conceptual schema of the ARGO-to-NetCDF conversion workflow.

In this paper, we present a workflow to convert ARGO observation data into a standard raster file. This workflow has been implemented in the context of a research e-Infrastructure with the aim to enhance the structure and re-usability of the ARGO data. We use an Open Science approach where all the standardized data are published in a Virtual Research Environment along with standardized metadata. The same conversion workflow is available as a Web service respecting the standard OGC Web Processing Service (*WPS*) and keeps track of the *provenance* of the data conversion process that allows reconstructing the processing history. This service was developed both to process all the historical ARGO data and to convert them as soon as new data are available. Our workflow transforms the ARGO unstructured data into NetCDF *Grid-feature* files. NetCDF is a self-describing, machine-independent data format meant to represent and store array-oriented data. It allows including information about the data in the file itself as attributes, and thus allows creating complex objects that do not need any external reference to be fully understood and thus re-usable and portable. The NetCDF Grid format assigns environmental variables values to a coordinate system with defined resolutions for longitude, latitude, depth/altitude, and time. Differently from the Point format, Grid is a raster format widely used by many communities and research institutions because many models/tools/libraries written in a large variety of programming languages are natively able to visualize, manipulate, and process this format. Our transformation workflow was implemented in *R*, thanks to availability of libraries to easily manipulate NetCDF files (e.g. ncdf4), and goes through the following steps:

1. Load a monthly-observation dataset in memory using *R*-specific importing functions optimized for large tables;
2. Represent metadata information, i.e. variable names and descriptions, units of measure, time instants, global parameters etc., in compliance with the Climate and Forecast (CF) standard vocabulary;
3. Represent all depth variables values in meters, using different conversion sub-routines depending on the original unit measure (e.g. pressures in dbar);
4. Generate a 3D grid with 10 logarithmic-divided depth intervals and 0.5° longitude-latitude resolution grids associated to each depth interval;
5. Clamp observation values to this 3D grid and associate values averages to each cell;
6. Create one NetCDF-CF file for every clamped variable.

This workflow is data-, memory-, and computing-intensive. Thus, in order to process the large amount of ARGO data we used a parallel processing and Cloud computing system named *DataMiner*[1] offered by the D4Science e-Infrastructure (www.d4science.org). DataMiner is an open source computational system that allows using Map-Reduce coupled with multi-core processing in scripts and programs written in a large variety of programming languages. An importing tool (SAI[2]) facilitates both software integration and use of distributed computing, and automatically generates a Web interface for the integrated software. We configured our workflow in order (i) to parallelise point 4 on several cores of one (virtual) machine in the DataMiner system, and (ii) to make each input ARGO CSV file processed by one different machine in the Cloud. With respect to other Cloud computing platforms, DataMiner publishes and describes the hosted processes under WPS and produces an XML provenance information file for all the executed processes in the Prov-O ontological format. Our workflow was published as a WPS service through DataMiner. It accepts one ARGO CSV-file as input and produces one NetCDF file for each variable included in the input file. Internally, the workflow uses the ARGO JSON file to retrieve metadata information. Another advantage of using DataMiner is that it interoperates with the other services of the D4Science e-infrastructure, i.e. (i) a distributed storage system for Big Data (the *Workspace*), (ii) data visualization, browsing, manipulation, and access services, (iii) social networking and collaborative spaces, (iv) security, authorization, and accounting services. D4Science supports Virtual Research Environments (VREs), i.e. online environments that foster collaboration between users and regulate users' access to data and services. We published our workflow in the *ScalableDataMining* VRE (accessible at services.d4science.org/web/scalabledatamining/), which grants free access to a DataMiner computing cluster made up of 20 Ubuntu machines with 32GB RAM, 16 virtual cores for single-machine parallelized processing, and 1TB of distributed and high-availability storage wherein the NetCDF files are uploaded after the computations. Through our workflow, the VRE users can process other ARGO data and optionally share the NetCDF files between them. Further, the VRE services allow accessing and visualising these files via OPeNDAP, and retrieving their metadata in ISO-19139 format[3]. In the ScalableDataMining VRE our workflow required ~15 hours to process all the ~200GB of ARGO data and produced ~120GB of NetCDF files, with provenance information associated to each file. These files were eventually published in the VRE catalogue[3] and made accessible through another VRE (the *BiodiversityLab*) that collects people interested in these data for various modelling applications.

Overall, our approach fosters re-usability of the data and goes towards Open Science and the free sharing of results and processes. It allows a user to add specific terms as variable- and global-attributes in order to connect a given dataset to domain-specific ontologies and thus to make it more understandable for certain communities of practice. This also allows using ARGO data in many

[1] Coro, G., Panichi, G., Scarponi, P., & Pagano, P. (2017). Cloud computing in a distributed e-infrastructure using the web processing service standard. Concurrency and Computation: Practice and Experience, 29(18). [2] Coro, G., Panichi, G., & Pagano, P. (2016). A Web application to publish R scripts as-a-Service on a Cloud computing platform. Bollettino di Geofisica Teorica ed Applicata. 57, 51-53. Proceedings of IMDIS 2016. [3] thredds.d4science.org/thredds/catalog/public/netcdf/Argo/catalog.html services.d4science.org/group/biodiversitylab/geo-visualisation

experiments, for example to feed ecological models and geospatial interpolation services of other e-Infrastructures (e.g. the SeaDataNet DIVA service, www.seadatanet.org/Software/DIVA).