

EGU23-3639, updated on 24 Feb 2024

<https://doi.org/10.5194/egusphere-egu23-3639>

EGU General Assembly 2023

© Author(s) 2024. This work is distributed under the Creative Commons Attribution 4.0 License.



## Data Proximate Computation; Multi-cloud approach on European Weather Cloud and Amazon Web Services

**Armagan Karatosun**<sup>1</sup>, Michael Grant<sup>2</sup>, Vasileios Baousis<sup>1</sup>, Duncan McGregor<sup>3</sup>, Richard Care<sup>3</sup>, John Nolan<sup>3</sup>, and Roope Tervo<sup>2</sup>

<sup>1</sup>European Centre for Medium-Range Weather Forecasts, Computing Department, Bonn, Germany

(armagan.karatosun@ecmwf.int)

<sup>2</sup>European Organisation for the Exploitation of Meteorological Satellites, User Support and Climate Services, Darmstadt, Germany (michael.grant@eumetsat.int)

<sup>3</sup>The Meteorological Office, Met Office Operations Centre, Exeter, Devon (john.nolan@metoffice.gov.uk)

Although utilizing the cloud infrastructure for big data processing algorithms is increasingly common, the challenges of utilizing cloud infrastructures efficiently and effectively are often underestimated. This is especially true in multi-cloud scenarios where data are available only on a subset of the participating clouds. In this study, we have iteratively developed a solution enabling efficient access to ECMWF's Numerical Weather Prediction (NWP) and EUMETSAT's satellite data on the European Weather Cloud [1], in combination with UK Met Office assets in Amazon Web Services (AWS), in order to provide a common template for multi-cloud processing solutions in meteorological application development and operations in Europe.

Dask [2] was chosen as the computing framework due to its widespread use in the meteorological community, its ability to automatically spread processing, and its flexibility in changing how workloads are distributed across physical or virtualized infrastructures while maintaining scalability. However, the techniques used here are generally applicable to other frameworks. The primary limitation in using Dask is that all nodes should be able to intercommunicate freely, which is a serious limitation when nodes are distributed over multiple clouds. Although it is possible to route between multiple cloud environments over the Internet, this introduces considerable administrative work (firewalls, security) as well as networking complexities (e.g., due to extensive use of potentially-clashing private IP ranges and NAT in clouds, or cost for public IPs). Virtual Private Networks (VPNs) can hide these issues, but many use a hub-and-spokes model, meaning that communications between workers pass through a central hub. By use of a mesh network VPN (WireGuard) between clusters using IPv6 private addressing, all these difficulties can be avoided, in addition to providing a simplified network addressing scheme with extremely high scalability. Another challenge was to ensure the Dask worker nodes were aware of data locality, both in terms of placing work near data and in terms of minimizing transfers. Here, the UK Met Office's work on labeling resource pools (in this case, data) and linking scheduling decisions to labels was the key.

In summary, by adapting Dask's concept of resourcing [3] into resource pools [4], building an automated start-up process, and effectively utilizing self-configuring IPv6 VPN mesh networks, we

managed to provide a “cloud-native” transient model where all resources can be easily created and disposed of as needed. The resulting “throwaway” multi-cloud Dask framework is able to efficiently place processing on workers proximate to the data while minimizing necessary data traffic between clouds, thus achieving results more quickly and cheaper than naïve implementations, and with a simple, automated setup suitable for meteorological developers. The technical basis of this work was published on the Dask blog [5] but is covered more holistically here, particularly regarding the application side and challenges of developing cloud-native applications which can effectively utilize modern multi-cloud environments, with future applicability to distributed (e.g., Kubernetes) and serverless computing models.

#### References:

- [1] <https://www.europeanweather.cloud>
- [2] <https://www.dask.org>
- [3] <https://distributed.dask.org/en/stable/resources.html>
- [4] <https://github.com/gjoseph92/dask-worker-pools>
- [5] <https://blog.dask.org/2022/07/19/dask-multi-cloud>