ENVRI^PLUS DELIVERABLE

D7.4

PERFORMANCE OPTIMISATION SERVICES FOR ENVIRONMENTAL ESFRI PROJECTS: PROTOTYPE

WORK PACKAGE 7—DATA PROCESSING AND ANALYSIS

LEADING BENEFICIARY: UNIVERSITY OF AMSTERDAM

Author(s)	Beneficiary/Institution
Paul Martin, Spiros Koulouzis, Zhiming Zhao	University of Amsterdam (UvA)

Accepted by: Leonardo Candela

Deliverable type: DEMONSTRATOR

Dissemination level: PUBLIC

Deliverable due date: 31.10.2018/M42

Actual Date of Submission: 31.10.2018/M42

A document of the ENVRI^PLUS project—http://www.envriplus.eu
This document has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 654182.

Abstract

This document is part of the final deliverable for **Task 7.2, "Performance optimisation for big data sciences"**, which addresses optimisation of e-infrastructure within the ENVRIplus **Data for Science** theme.

The complete deliverable consists of the technologies developed within ENVRIplus based on the design vision and roadmap provided by its predecessor deliverable, **D7.3, "Performance optimisation for environmental RI projects: system design"**. This document reviews the technologies demonstrated in the project for this task, and provides the necessary information to retrieve and utilise these technologies. The three technology demonstrations described are the **Dynamic Real-time Infrastructure Planner (DRIP)**, a virtual overlay for **Information-Centric Networking** and a **Data Subscription Service**, all of which demonstrate optimisation of e-infrastructure based on data-driven requirements imposed by scientific applications.

Project internal reviewer(s)

Project internal reviewer(s)	Beneficiary/Institution
Yin Chen	EGI.eu
Barbara Magagna	Umweltbundesamt GmbH (EAA/LTER)

Document history

Date	Version
12/9/18	Initial version compiled.
30/9/18	Final edits pending review.
26/10/18	Revised version based on internal review.

Document amendment procedure

Amendments, comments and suggestions should be sent to Paul Martin (p.w.martin@uva.nl) or Zhiming Zhao (z.zhao@uva.nl).

Terminology

This deliverable uses terminology based on the ENVRI Reference Model [15], which is published online as an ontology:

`http://www.oil-e.net/ontology/envri-rm.owl`

A complete project glossary is provided online here:

`https://wiki.envri.eu/pages/viewpage.action?pageId=14452608`

Project summary

ENVRIplus is a Horizon 2020 project bringing together Environmental and Earth System Research Infrastructures, projects and networks together with technical specialist partners to create a more coherent, interdisciplinary and interoperable cluster of Environmental Research Infrastructures across Europe. It is driven by three overarching goals: 1) promoting cross-fertilization between infrastructures, 2) implementing innovative concepts and devices across RIs, and 3) facilitating research and innovation in the field of environment for an increasing number of users outside the RIs.

ENVRIplus aligns its activities to a core strategic plan where sharing multi-disciplinary expertise will be most effective. The project aims to improve Earth observation monitoring systems and strategies, including actions to improve harmonization and innovation, and generate common solutions to many shared information technology and data related challenges. It also seeks to harmonize policies for access and provide strategies for knowledge transfer amongst RIs. ENVRIplus develops guidelines to enhance transdisciplinary use of data and data-products supported by applied use-cases involving RIs from different domains. The project coordinates actions to improve communication and cooperation, addressing Environmental RIs at all levels, from management to end-users, implementing RI-staff exchange programs, generating material for RI personnel, and proposing common strategic developments and actions for enhancing services to users and evaluating the socio-economic impacts.

ENVRIplus is expected to facilitate structuration and improve quality of services offered both within single RIs and at the pan-RI level. It promotes efficient and multi-disciplinary research offering new opportunities to users, new tools to RI managers and new communication strategies for environmental RI communities. The resulting solutions, services and other project outcomes are made available to all environmental RI initiatives, thus contributing to the development of a coherent European RI ecosystem.

Table of contents

1 Introduction

The integration of data from a diversity of sources is a necessary component of system-level science [5], that is science that considers complex systems with multiple intertwined processes. The ESFRI and other RI landmark projects[1] represented by ENVRIplus are intended to support such integration across the environmental and earth sciences. Beyond the larger consolidated data centres however, there is a great variety of long-tail data in small datasets scattered across many sites, and the velocity of data gathering (at all scales) continues to increase. Investigations of research data require not only access to those distributed data sources, but also reliable access to e-infrastructure onto which that data can be staged and processed. To ensure that this access is available to as broad a community of researchers as possible, there are a number of important issues to resolve, many of which are directly addressed by ENVRIplus, including administrative and legal issues. Here however we concern ourselves with some of the *technical* issues as they pertain to the needs of different RI use-case scenarios: for example how to efficiently transport large datasets over the network, how to efficiently provision (virtual) infrastructure to support complex workflows, how to steer data processing activities at runtime, and when best to preserve or cache intermediate data that might serve to reduce unnecessary process replication or data traffic.

This report is concerned with reviewing and providing links to the technologies developed and demonstrated to address the topic of 'optimisation' as part of the second deliverable of Task 7.2 of the ENVRIplus project.

1.1 Relation to task

Within ENVRIplus, **Task 7.2 "Performance optimisation for big data sciences"** is concerned with characterising the availability and scope of data and computational resources provided for and by research infrastructures (RIs), and with the development of tools and services for optimising the performance of experiments and other data-driven interactions executed via those RIs. More precisely, the task's scope encompasses investigations into the optimisation of data, resource and service configuration on underlying e-infrastructure, and the scheduling of tasks on that e-infrastructure. This task is heavily embedded in the cross-cutting activities within the ENVRIplus "Data for Science" theme, especially those of Work Package 5 "Reference model guided RI design". It works in tandem with the concurrent activities in other work packages in the theme, particularly its sister task, Task 7.1 "Interoperable Data Processing, Monitoring and Diagnosis".

Three technology demonstrations are described in this document: the **Dynamic Real-time Infrastructure Planner (DRIP)**, a virtual overlay for **Information-Centric Networking** and a **Data Subscription Service**, all of which demonstrate optimisation of e-infrastructure based on data-driven requirements imposed by scientific applications. These technologies collectively address the following actions ascribed to Task 7.2 by the ENVRIplus description of work:

- Provide an effective mapping between application-level and infrastructure-level requirements, and identify the conditions under which tools and services can be deployed—DRIP allows for the scheduling of e-infrastructure resources based on deadline requirements imposed on distributed application workflows; we have examined requirements of scientific application on e-infrastructure in publications such as [9] and [11].
- Prototype components to assist with the optimisation of data movement and processing—DRIP provides a flexible platform for the planning, deployment and execution of data processing applications on Cloud-based e-infrastructure.
- Investigate the use of tools for large-scale data analysis in the context of e-infrastructure—expanding on the technology survey of Deliverable 7.3, we have investigated the use of technologies such as ICN and built a pipeline for data subscription over Cloud infrastructure.

1.2 Relation to previous deliverable

The previous deliverable of this task, **D7.3, "Performance optimisation for environmental RI projects: system design"** [13] provided a vision and roadmap which was then used to guide the

[1] http://ec.europa.eu/research/esfri/

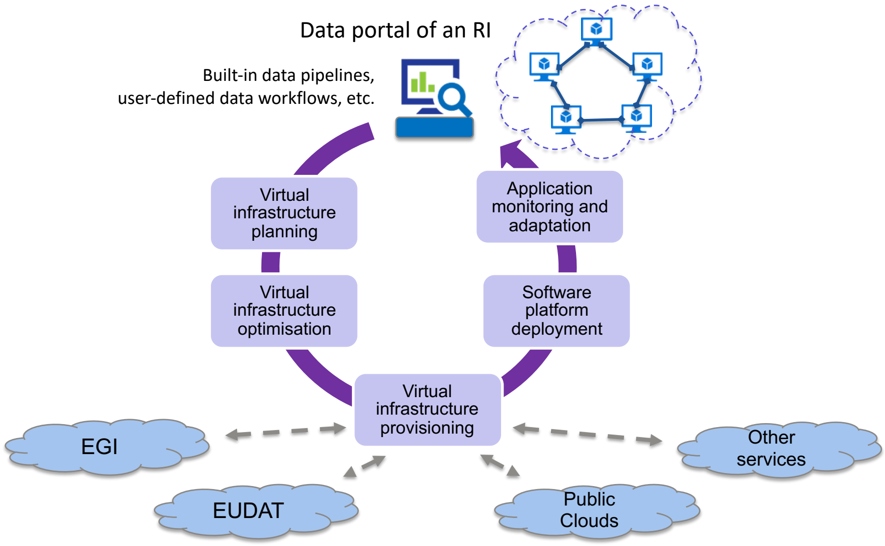


Figure 1: Optimisation microservices in the context of research infrastructure.

development of prototype technologies to address the optimisation of e-infrastructure for RIs in terms of efficient allocation of virtual resources and networks, and efficient data movement. Figure 1 shows the overall concept: the development of microservices for planning, provisioning, deploying, *etc.* application components on e-infrastructure provided via intiatives such as EGI or EUDAT.

Broadly speaking, the DRIP platform developed within this task fulfils the vision articulated in the previous deliverable: it provides a number of components (microservices) for flexible deployment of distributed applications on Cloud-style e-infrastructure which can be used independently, or together via a simple RESTful interface. The viability of DRIP and of other technologies also surveyed in [13] (such as Apache Spark) have also been demonstrated to the community via the Euro-Argo Data Subscription Service use-case, which was also shown at the ENVRIplus project mid-term review.

1.3 Layout

The remainder of this document reviews the demonstrators presented to the ENVRI community in the context of the optimisation task:

Section 2 "Dynamic Real-time Infrastructure Planner" describes the eponymous optimisation microservice suite DRIP.

Section 3 "Information-centric networking" describes the demonstrator for the use of ICN for data routing based on persistent identifiers of scientific datasets.

Section 4 "Data subscription case" describes the data subscription use-case constructed by partners in the task, which uses DRIP and the e-infrastructure provided by EUDAT and EGI to optimise data transfer. The use-case is also summarised in Deliverable 9.2 [3].

All technologies are available via specific links (published in their respective sections) and are part of the ENVRI service portfolio, which can be browsed at:

`https://ckan-d4s.d4science.org/organization/envriplusdata4science`

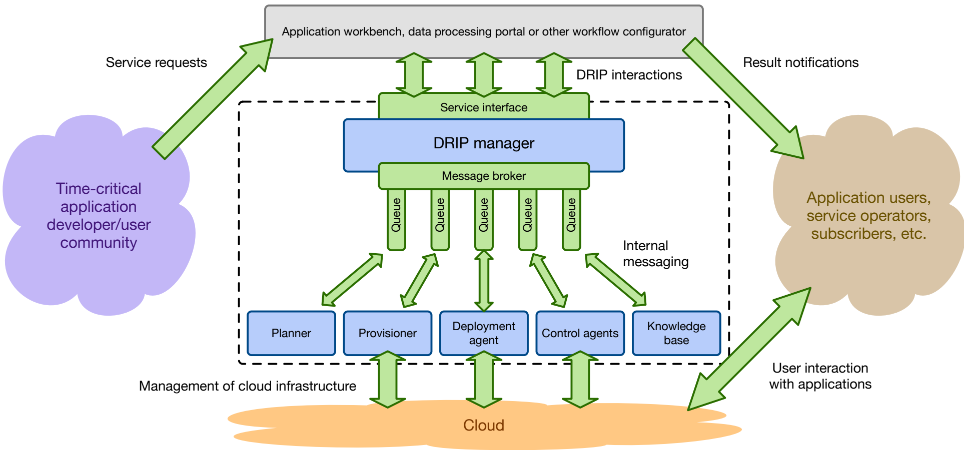


Figure 2: How services provided by DRIP are invoked by RI services to provide downstream services to users.

2 Dynamic Real-time Infrastructure Planner

The *Dynamic Real-time Infrastructure Planner* (DRIP) is a microservice suite for planning and provisioning networks of virtual machines and then deploying distributed applications across those networks; DRIP then manages the virtual infrastructure during run-time based on time-critical constraints defined with the application workflow [21]. DRIP provides an engine for automating all these procedures by making use of pluggable microservices orchestrated via a single manager component behind a RESTful Web API, thus providing a prototype of the e-infrastructure optimisation architecture described in [13] and shown in Figure 1 in the previous section.

2.1 Download link

DRIP is available online under the Apache 2.0 open source license at the following URL:

`https://github.com/QCAPI-DRIP/`

Documentation and tutorials can be found at:

`https://github.com/QCAPI-DRIP/DRIP-integration/wiki`

A video of DRIP in action (originally prepared for the H2020 SWITCH project[2]) can be seen at:

`https://www.youtube.com/watch?v=bSADnxgT8p8&t=3s`

2.2 Architecture

The DRIP services include a number of components, interacting via an internal message brokering service orchestrated by a single manager. These components and their interaction are shown in Figure 2. All components of DRIP under the control of the DRIP manager are designed to be independently replaceable, to allow for improved or alternative implementations of *e.g.* the planner or the provisioner. The types of service that can be integrated within DRIP, and their current implementations, are:

The DRIP manager. The manager is a Web service that allows DRIP functions to be invoked by external clients. Each request is directed to the appropriate component by the manager, which coordinates the individual components and scales them up if necessary.

Infrastructure planner. The current planner uses a partial critical path algorithm [1] optimised for workflows with multiple internal deadlines to produce efficient infrastructure topologies, selecting the most cost-effective virtual machines [18]. Potentially, multiple planners can be

[2] http://www.switch-project.eu/

attached to DRIP in order to manage different kinds of application workflow or infrastructure topology, exploiting technologies such as software-defined networking [12] to customise the network topology among VMs and optimally place controllers for the networked VMs [17].

Infrastructure provisioner. The provisioner is responsible for automating the provisioning of infrastructure plans produced by the planner(s) onto the underlying cloud or e-infrastructure. The current provisioner can decompose the infrastructure description and provision it across multiple data centres (possibly from different providers) with transparent network configuration [23].

Deployment agent. The deployment agent installs application components onto provisioned infrastructure. The current deployment agent is able to schedule the deployment sequence based on network bottlenecks, and maximise the fulfillment of deployment deadlines for all the cloud providers currently supported by the default DRIP provisioner [7].

Infrastructure control agents. These agents provide sets of APIs that DRIP can then provide to applications to control the scaling of containers or VMs and for adapting network flows or use itself in conjunction with a monitoring framework to automatically maintain the quality of service of the deployed application.

Knowledge base. To allow different service components to operate effectively, DRIP provides a simple knowledge base for storing information about user credentials, the types of resource offered by clouds or e-infrastructure, and other useful data that the DRIP manager or any other component can retrieve or contribute to.

The prototype of DRIP adopts industrial and community standards such as TOSCA[3] for descriptions of applications and their constraints and OCCI as its default provisioning interface, supporting the Amazon EC2[4], EGI FedCloud[5] and ExoGeni[6] Clouds. The deployment agent can deploy overlay Docker clusters using Docker Swarm[7] or Kubernetes[8]. RabbitMQ[9] is used for internal message broker. All DRIP software is open source.

2.3 How DRIP works

To function, DRIP requires:

- An application description from the developer, identifying the specific components to be deployed on the provisioned infrastructure and defining the dependencies between components that describe the application workflow and the time-critical constraints that apply to it.
- Information about the infrastructure resources (e.g. VM types and network bandwidth) obtained from the cloud providers, and their performance hosting specific applications (e.g. provided by a cloud discovery and profiling service [4]).

The application topology is currently described using TOSCA and must be part of the request made to DRIP. When a planning request comes, the manager will direct the request to the *infrastructure planner* to generate a plan, which can be sent back to user for further confirmation. If the constraints cannot be satisfied the planner informs the user that a plan cannot be generated. The DRIP manager stores necessary cloud credentials on behalf of the user. The *provisioning agent* can provision the virtual infrastructure via interfaces offered by the Cloud providers. Once this has finished, the *deployment agent* will deploy all necessary components onto the provisioned infrastructure from designated repositories and set up the control interfaces needed for runtime control of application and infrastructure.

[3]https://www.oasis-open.org/committees/tosca/
[4]https://aws.amazon.com/cn/ec2
[5]https://www.egi.eu/federation/egi-federated-cloud/
[6]http://www.exogeni.net/
[7]https://github.com/docker/swarm
[8]https://kubernetes.io/
[9]https://www.rabbitmq.com/

3 Information-Centric Networking

Persistent identifiers (PIDs) are an important mechanism for RIs to facilitate the long-term identification, publishing and citation of scientific data objects. As the amount of stored data grows rapidly, RIs face challenges in accommodating the increased volume and frequency of transfer requests however. Data movement is frequent during processing and analyzing of data objects, in particular when an application uses multiple resources across multiple sites.

We investigated the mapping of PIDs onto the Named Data Networking (NDN) [19] naming schema, an implementation of Information-Centric Networking (ICN). We also investigated the behaviour of different caching mechanisms and based on their feasibility proposed an agenda for promoting NDN as a sharing and distribution service for data infrastructure [11].

3.1 Download link

The experimental code for routing datasets based on the Named Data Networking implementation of ICN is available online at the following URL:

`https://github.com/skoulouzis/ndn-with-docker`

3.2 Motivation

Delivering large data volumes over IP networks causes severe delays to the execution of scientific workflows [6]. This has led the creation of technologies and applications such as MTCP [20], GridFTP [2], and FDT [14] and light-paths using software-defined networking [10]. All the aforementioned solutions are based on IP networks however, which present limitations when delivering large volumes of data to multiple consumers. That is because the data-centric overlay needed by data-incentive applications is poorly matched to the Internet's point-to-point underlay.

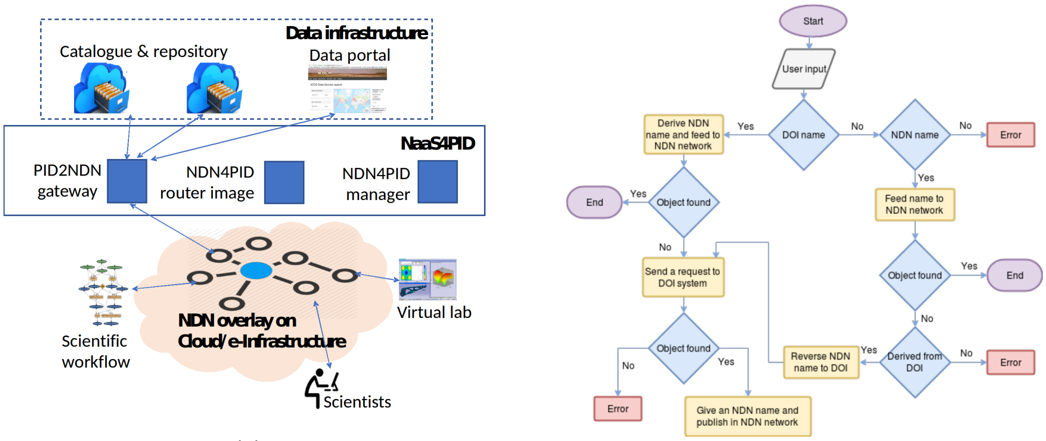
Information-Centric Networking (ICN) is a promising approach for networking, and is founded upon the idea that most users are interested in accessing data content, irrespective of its location. Instead of having an end-to-end communication model between network nodes, ICN infrastructure routes data objects between sources and destinations based on the unique names of the data objects, and caches data objects on intermediate locations in order to achieve efficient and reliable distribution of data content. ICN offers a natural architecture for transferring big data objects; of particular interest to RIs with large datasets is whether it is feasible to use the persistent identifiers already now being used to uniquely and globally identify datasets as a means to find and route data in ICN-based networks. At present, using ICN to deliver PID-based data objects still faces technical challenges: not only are the current schemas of identifiers for PIDs and ICN objects not fully interoperable, but also the resolution and handling services of ICN do not fit well with the current PID definition. These problems motivated us to investigate how to use available ICN implementations to deliver Big Data objects based on their PIDs in a more effective manner.

3.3 Architecture

An NDN-as-a-service for PID data objects (NaaS4PID) was proposed in [11] to distribute digital objects using NDN. NaaS4PID provides mapping services between PID and NDN naming schema, and message control mechanisms for inter-parsing and handling PIDs in NDN. Figure 3a presents the basic architecture.

NaaS4PID has three key components: the PID2NDN gateway; the NDN4PID router image; and the NDN4PID manager. The first is primarily responsible for resolving PIDs as NDN names. The NDN4PID router image is an NDN node that implements a virtualized NDN router. Finally, the NDN4PID manager automates the management of the NDN overlay on Cloud or e-infrastructure. The flow chart of the proposed fetching of digital objects (for the example of DOIs) is shown in Figure 3b.

When the requested name is a DOI, the corresponding NDN name would be first derived from it and an *interest packet* with that name would be sent to the NDN network; if the object has been requested before and is still available cached in the NDN network, then it can be retrieved as a *data*



(a) (b)

Figure 3: (a) NaaS4PID architecture; (b) Fetching digital objects via an NDN network.

packet to the user. Otherwise a request is sent to the DOI system to fetch the object; a new NDN name is derived from it if necessary and it is published to the NDN network. NDN names can be used directly by clients, in which case it should be possible to locate the data object referred to in the NDN network.

The NDN4PID router image accommodates data requests and should be deployed next to scientific applications to take advantage of NDN capabilities. To optimize the data distribution, application-level knowledge, *e.g.* objects' sizes, number and ordering, is needed for the caching and routing of NDN. Such information may be obtainable from a metadata catalogue. During the execution of a scientific workflow, such an application may also need more data objects, *e.g.* for tuning the simulation or for comparison purposes. These data objects may be published by different producers, located in different locations or have different sizes. This requires investigation of different caching strategies.

The NDN4PID manager automates the management of the NDN overlay in Cloud e-infrastructures such as EGI. It can dynamically set up a virtual NDN overlay, deploy new routers and manage the NDN, and in experiments is powered by DRIP (see Section 2).

4 Data subscription service

The Euro-Argo RI prototyped a *subscription service* for the data gathered from the Argo float network. In contrast to simply providing collected data freely for download but requiring researchers to manually monitor the core Argo dataset for updates, researchers are instead being allowed to subscribe to specific subsets of Argo data and have updates pushed to their own cloud storage, thus streamlining data delivery and accelerating data science workflows involving those data.

The service has been demonstrated to the ENVRI community, and was presented at the project's mid-term review. A video detailing the use-case is available at:

`https://youtu.be/PKU_JcmSskw`

4.1 Service concept

A functional depiction of the Euro-Argo data subscription service is illustrated by Figure 4.

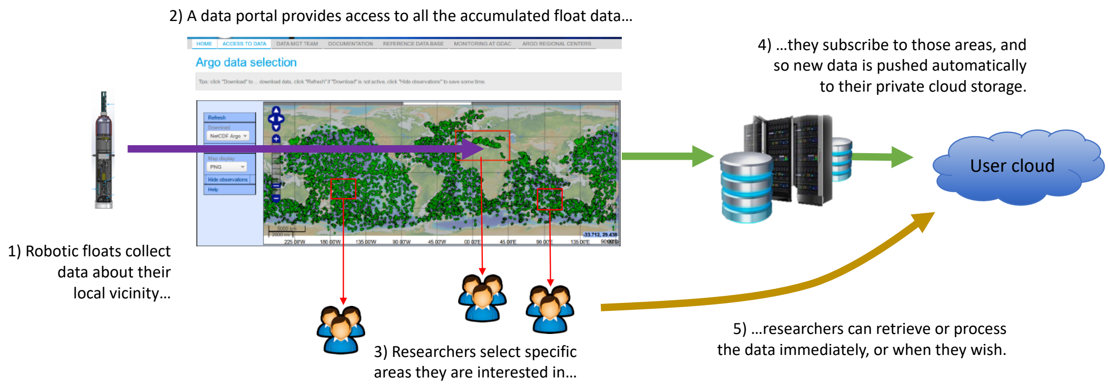


Figure 4: The data subscription scenario is one where researchers can subscribe to the specific data they are interested in (e.g. marine data from floats in the Mediterranean) via a simple portal, and have all updates pushed to their workspaces periodically.

Data subscription is a good example of the kind of scalable, customised data service that RIs are now looking at as a way to better serve their communities, and which requires some degree of optimisation of the underlying distributed infrastructure. It is also an example of a time-critical service; subscriptions should be fulfilled on a schedule (possibly mixed for different products, leading to sporadic peaks of activity as schedules for different products coincide), but different products may require different degrees of processing at different times and place differing levels of load on the network to deliver to subscribers. A typical subscription task is made up of a set of inputs:

1. An *area* expressed as a bounding box (geospatial data being very common in environmental and earth science).
2. A *time range* (typically investigators will want the most recent data, but updates to past readings due to quality control or restoration of missing data may also be of interest).
3. A list of *parameters* required in the data products (e.g. temperature or salinity; in advanced cases this may be a derivative parameter which must itself be computed from some base parameters).
4. Optionally, a deadline (otherwise, a standard update schedule will be applied; deadlines might be expressed in terms of time since last update, or simply be a regular recurring window for delivery).

Such a data subscription service serves both end-users and application workflows for which the retrieval of subscribed data is a key input. Often these workflows require specific data to be delivered within a specific time window and often have firm or soft real-time requirements. The type of real-time requirement is specified by the developer.

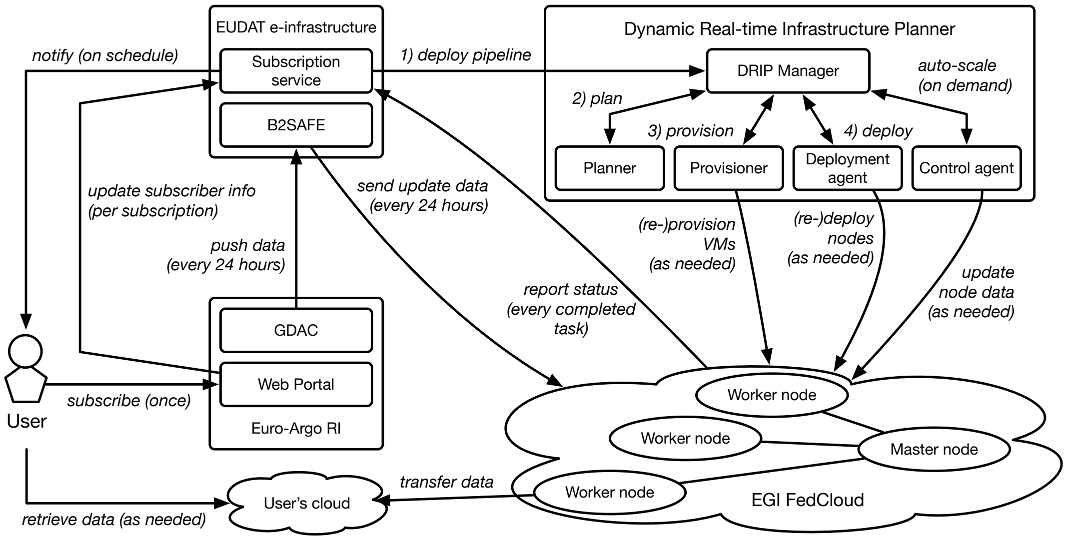


Figure 5: The architecture of the Euro-Argo data subscription service with DRIP. The subscription service invokes DRIP to plan, provision for and deploy the subscription data processing pipeline. Subscriptions and processing are event-driven, triggered by updates pushed to the B2SAFE data store. The deployment is scaled with demand.

As the volume of subscriptions and the customisability of subscriptions increases, so too does the pressure on the underlying infrastructure providing the data, the bandwidth for transport and the processing. At the same time however, there will be periods of low activity between rounds of updates. Thus we need a scalable infrastructure to support the data subscription processing pipeline, so as to not unnecessarily tie up resources while still permitting acceptable quality of service during peak periods.

Beyond the parameters of this particular scenario, we also should consider how we manage the e-infrastructure used by RIs where there may be multiple data subscription pipelines for different kinds of data, or indeed how to easily configure and deploy new pipelines should other RIs want to replicate the Euro-Argo system for their own scientific datasets.

4.2 Service prototype

We prototyped the data subscription service based on the architecture depicted in Figure 5, using resources from EUDAT and EGI FedCloud. In this case, EUDAT provides services for storage and data transfer, while EGI FedCloud provides the services for the computing of data products for each subscription.

The data subscription service scenario thus involves the following basic components:

1. A data selection portal serving as the front-end.
2. The global data assembly centre of Euro-Argo, providing the source research dataset.
3. B2SAFE storage provided by EUDAT.
4. A deployment of DRIP (see Section 2) deployed within EGI FedCloud.
5. EGI FedCloud virtual resources, forming the fundamental infrastructure for data processing and transportation.
6. The subscription service itself (which maintains the subscriptions defined via the data selection portal).

Users interact with the subscription service via a portal, registering to receive updates for specific areas and time ranges for selected parameters such as temperature, salinity, and oxygen levels. The global data assembly centre (GDAC) of Euro-Argo receives new datasets from regional centres and pushes them to the B2SAFE data service. The subscription service itself maintains records of subscriptions including selected parameters and associated actions. The role of DRIP then is to plan and provision customised infrastructure dynamically with demand, and to deploy, scale and control the data filtering application to be hosted on that infrastructure. EGI FedCloud provides actual cloud resources provisioned by DRIP.

The application itself is composed of a master node and a set of worker nodes. The *master node* uses a monitoring process that tracks specified metrics and interacts with the DRIP controller, which can scale out workers on demand. The master is also responsible for partitioning input parameters and distributing them to workers for parallel execution then re-combination. Partitioning input parameters should provide faster execution due to increased speed-up. The *workers* perform the actual query on the dataset based on the partitioned input parameters provided by the master node.

When new data is available to the GDAC, it pushes them to the B2SAFE service, triggering a notification to the subscription service, which consequently initiates actions on the new data. If the application is not deployed to FedCloud already, then DRIP provisions the necessary VMs and network so that the application may be deployed. Next, the deployment agent installs all the necessary dependencies along with the application including configurations to access on the Argo data. The subscription service signals to the application master node the availability of the input parameters to be processed, whereupon it partitions the input tasks into sub-tasks and distributes them to the workers. If the input parameters include deadlines then the master will prioritise them accordingly. The monitoring process keeps track of each running task and passes that information to the DRIP controller. If the programmed threshold is passed, then the controller will request more resources from the provisioner. Finally, the results of each task are pushed back to the B2SAFE service triggering a notification to the subscription service, after which it notifies the user.

5 Summary

The optimisation of e-infrastructure for scientific applications has become especially important with the emergence of the European Open Science Cloud (EOSC) initiative [16] during the lifespan of ENVRIplus. It is clear that effective use of common e-infrastructure and scientific Cloud are critical to EOSC, and thus we need tools and services for planning and provisioning dedicated infrastructure for a variety of different data processing workflows.

As the ENVRI community prepares to enter its next phase of development with the ENVRI-FAIR project, optimisation will continue to play an important part; a significant portion of the effort in ENVRI-FAIR is dedicated to the deployment and operation of common services (developed across the ENVRI and ENVRIplus projects) on behalf of each of the four major domains of environmental ESFRI research infrastructures (atmosphere, marine, solid earth and ecosystem/biodiversity). Once again, the tools and insights garnered during this task should prove invaluable to realising these services in practice.

The three technology contributions represent different aspects of optimisation at different levels, from 'deep infrastructure' (the use of ICN) to integrated data processing pipelines (the data subscription service). All present compelling avenues for future research and development.

- The investigation of Information-Centric Networking (ICN), for example the work on the use of PIDs within NDN networks described in this report, represents a new kind of data-oriented navigation of online resources that could have significant future impact on how scientific data is provided to researchers by RIs in future; this requires however future research into how to manage different versions of data, maintain data provenance, and ensure proper accounting and attribution of the use of datasets.
- The Dynamic Real-time Infrastructure Planner (DRIP) provide an architecture supporting the development of small, dedicated microservices for use on behalf of RIs and researchers who wish to deploy data processing or scientific workflows on Cloud-based e-infrastructure. DRIP continues to benefit from active development, with refinements to provisioning [22] and deployment scheduling [8] functionalities being examples of recent developments.
- The Data Subscription Service represents a practical use-case for data optimisation that has clear applicability to RIs in general; the ability for researchers to subscribe to datasets, to define the frequency of updates and the parameters they are specifically interested in, coupled with a robust infrastructure that can support many requests in parallel, would notably expedite the use of RI data by researchers in general. The future possibility of subscriptions that join data collected from multiple RIs into a single product is also of great interest.

The three main technology contributions described above represent the main 'dedicated' technical outcomes of the optimisation task in ENVRIplus; it should be remembered however that ENVRIplus is a highly interconnected project, with all project partners involved in multiple strands of activity in the project. Thus, members of the optimisation task have contributed to many other parts of the project and carried forward many of the ideas generated within the task into those other activities: the scope of this promulgation can be seen in the ENVRI service portfolio[10], in the review of use-cases provided by Deliverable 9.2 [3] and in this report's predecessor, Deliverable 7.3 [13].

[10] https://ckan-d4s.d4science.org/organization/envriplusdata4science

References

[1] Saeid Abrishami, Mahmoud Naghibzadeh, and Dick HJ Epema. Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds. *Future Generation Computer Systems*, 29(1):158–169, 2013.

[2] John Bresnahan, Michael Link, Rajkumar Kettimuthu, Dan Fraser, Ian Foster, et al. Gridftp pipelining. In *Proceedings of the 2007 TeraGrid Conference*, 2007.

[3] Yin Chen, Ingemar Häggström, Justin Buck, Markus Stocker, Thierry Carval, Domenico Vitale, Robrt Huber, Margareta Hellström, Leonardo Candela, and Florian Haslinger. *D9.2 Service deployment in computing and data e-infrastructures, version 2*. H2020 ENVRIplus Project (http://www.envriplus.eu), 2018.

[4] Olaf Elzinga, Spiros Koulouzis, Arie Taal, Junchao Wang, Yang Hu, Huan Zhou, Paul Martin, Cees de Laat, and Zhiming Zhao. Automatic collector for dynamic cloud performance information. In *Networking, Architecture, and Storage (NAS), 2017 International Conference on*, pages 1–6. IEEE, 2017.

[5] Ian Foster and Carl Kesselman. Scaling system-level science: Scientific exploration and it implications. *Computer*, 39(11):31–39, 2006.

[6] Christina Hoffa, Gaurang Mehta, Tim Freeman, Ewa Deelman, Kate Keahey, Bruce Berriman, and John Good. On the use of cloud computing for scientific workflows. In *eScience, 2008. eScience'08. IEEE Fourth International Conference on*, pages 640–645. IEEE, 2008.

[7] Yang Hu, Junchao Wang, Huan Zhou, Paul Martin, Arie Taal, Cees de Laat, and Zhiming Zhao. Deadline-aware deployment for time critical applications in clouds. In *European Conference on Parallel Processing*, pages 345–357. Springer, 2017.

[8] Yang Hu, Huan Zhou, Cees de Laat, and Zhiming Zhao. Ecsched: Efficient container scheduling on heterogeneous clusters. In *European Conference on Parallel Processing*, pages 365–377. Springer, 2018.

[9] S Koulouzis, P Martin, T Carval, B Grenier, G Judeau, J Wang, H Zhou, C de Laat, and Z Zhao. Seamless infrastructure customisation and performance optimisation for time-critical services in data infrastructures. In *Proceedings of the 8th International Workshop on Data-Intensive Computing in the Clouds, ACM SIGHPC, in IEEE Supercomputing*, 2017.

[10] Spiros Koulouzis, Adam Belloum, Marian Bubak, Zhiming Zhao, Miroslav Živković, and Cees de Laat. SDN-aware federation of distributed data. *Future Generation Computer Systems*, 56:64–76, 2016.

[11] Spiros Koulouzis, Rahaf Mousa, Andreas Karakannas, Cees de Laat, and Zhiming Zhao. Information centric networking for sharing and accessing digital objects with persistent identifiers on data infrastructures. In *2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. IEEE, 2018.

[12] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76, 2015.

[13] Paul Martin, Zhiming Zhao, Markus Stocker, Robert Huber, Jani Heikkinen, and Aleksi Kallio. *D7.3 Performance optimisation for environmental RI projects: system design*. H2020 ENVRIplus Project (http://www.envriplus.eu), 2017.

[14] Zdenek Maxa, Badar Ahmed, Dorian Kcira, Iosif Legrand, Azher Mughal, Michael Thomas, and Ramiro Voicu. Powering physics data transfers with fdt. *Journal of Physics: Conference Series*, 331(5), 2011.

[15] Abraham Nieva de la Hidalga, Barbara Magagna, Markus Stocker, Alex Hardisty, Paul Martin, Zhiming Zhao, Malcolm Atkinson, and Keith Jeffery. The ENVRI Reference Model (ENVRI RM) version 2.2, November 2017.

[16] The European Commission. *Realising the European Open Science Cloud*. The European Commission, 2016.

[17] Junchao Wang, Cees de Laat, and Zhiming Zhao. QoS-aware virtual SDN network planning. In *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on*, pages 644–647. IEEE, 2017.

[18] Junchao Wang, Arie Taal, Paul Martin, Yang Hu, Huan Zhou, Jianmin Pang, Cees de Laat, and Zhiming Zhao. Planning virtual infrastructures for time critical applications with multiple deadline constraints. *Future Generation Computer Systems*, 75:365–375, 2017.

[19] Lan Wang, AKMM Hoque, Cheng Yi, Adam Alyyan, and Beichuan Zhang. OSPFN: An OSPF based routing protocol for named data networking. Technical report, Technical Report NDN-0003, 2012.

[20] Carey Williamson and Niklas Carlsson. On zipf models for probabilistic piece selection in p2p stored media streaming. In *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2013 IEEE 21st International Symposium on*, pages 161–171. IEEE, 2013.

[21] Zhiming Zhao, Paul Martin, Andrew Jones, Ian Taylor, Vlado Stankovski, Guadalupe Flores Salado, George Suciu, Alexandre Ulisses, and Cees de Laat. Developing, provisioning and controlling time critical applications in cloud. In *European Conference on Service-Oriented and Cloud Computing*, pages 169–174. Springer, 2017.

[22] Huan Zhou, Yang Hu, Jinshu Su, Cees de Laat, and Zhiming Zhao. Cloudsstorm: An application-driven framework to enhance the programmability and controllability of cloud virtual infrastructures. In *International Conference on Cloud Computing*, pages 265–280. Springer, 2018.

[23] Huan Zhou, Yang Hu, Junchao Wang, Paul Martin, Cees de Laat, and Zhiming Zhao. Fast and dynamic resource provisioning for quality critical cloud applications. In *2016 IEEE 19th International Symposium on Real-Time Distributed Computing (ISORC)*, pages 92–99. IEEE, 2016.