# Big data analytics for climate change and biodiversity in the EUBrazilCC federated cloud infrastructure

Fiore, S.; Mancini, M.; Elia, D.; Nassisi, P.; Vilar Brasileiro, F.; Blanquer, I.; Rufino, I.A.A.; Seijmonsbergen, A.C.; de Oliveira Galvao, C.; Perez Canhos, V.; Mariello, A.; Palazzo, C.; Nuzzo, A.; D'Anca, A.; Aloisio, G.

**DOI**
10.1145/2742854.2747282

**Publication date**
2015

**Document Version**
Final published version

**Published in**
CF '15

# Big data analytics for climate change and biodiversity in the EUBrazilCC federated cloud infrastructure

Sandro Fiore
Euro-Mediterranean Center on Climate Change
Lecce, Italy
sandro.fiore@cmcc.it

Marco Mancini
Euro-Mediterranean Center on Climate Change
Lecce, Italy
marco.mancini@cmcc.it

Donatello Elia
Euro-Mediterranean Center on Climate Change
Lecce, Italy
donatello.elia@cmcc.it

Paola Nassisi
Euro-Mediterranean Center on Climate Change
Lecce, Italy
paola.nassisi@cmcc.it

Francisco Vilar Brasileiro
Universidade Federal de Campina Grande
Campina Grande, PB, Brasil
fubica@dsc.ufcg.edu.br

Ignacio Blanquer
Universitat Politecnica de Valencia
Valencia, Spain
iblanque@dsic.upv.es

## ABSTRACT

The analysis of large volumes of data is key for knowledge discovery in several scientific domains such as climate, astrophysics, life sciences among others. It requires a large set of computational and storage resources, as well as flexible and efficient software solutions able to dynamically exploit the available infrastructure and address issues related to data volume, distribution, velocity and heterogeneity. This paper presents a data-driven and cloud-based use case implemented in the context of the EUBrazilCC project for the analysis of climate change and biodiversity data. The use case architecture and main components, as well as a Platform as a Service (PaaS) framework for big data analytics named PDAS, together with its elastic deployment in the EUBrazilCC federated cloud infrastructure are presented and discussed in detail.

## Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Scientific databases; C.2.4 [**Distributed Systems**]: Distributed applications

## Keywords

Cloud computing; scientific data management; big data analytics; federated clouds.

## 1. INTRODUCTION

Flexible and elastic resource provisioning, resource federation, application porting and big data analysis frameworks are key components to build sustainable ecosystems for scientific knowledge discovery [1]. The EUBrazilCloudConnect project [2] (which relies on a close collaboration between a strong partnership of European and Brazilian excellence) is a preliminary step towards providing a user-centric environment for the European and Brazilian research communities to test the execution of challenging scientific applications.
EUBrazilCC is exploiting and coordinating, in a 2-year project, a heterogeneous e-Infrastructure over a wide geographic cross-Atlantic area involving both computer clusters and private clouds.
The project addresses the scientific challenges of three multidisciplinary and highly complementary scenarios, covering epidemiology, health, biodiversity, natural resources and climate change. The proposed scientific scenarios require access to the project e-infrastructure to run complex workflow pipelines as well as access to heterogeneous and large datasets.
This work presents the results related to the use case on climate change and biodiversity, the scientific challenges and the cloud-based solution related to this data-driven use case.
The remainder of this work is organised as follows: Section 2 presents the use case on climate change and biodiversity with a specific application-domain focus. Section 3 describes the use case architecture, whereas Section 4 presents the related infrastructure, highlighting the security model, the infrastructure manager and federation layer, the Infrastructure as a Service (IaaS) layer, the Parallel Data Analytics Service (PDAS) and the Scientific Gateway for the end-user community. With regard to the PDAS, the paper provides an in-depth discussion highlighting the internal architecture, its role in the EUBrazilCC project, three deployment scenarios in a cloud environment, the client-side, and a real workflow related to a climate indicator. Section 5 describes the major impact and added value provided by the proposed solution. Finally, Section 6 draws the conclusions and discuss future work.

## 2. USE CASE ON CLIMATE CHANGE AND BIODIVERSITY

The EUBrazilCC use case on climate change and biodiversity is a data-driven use case, aiming at better understanding the interactions (status and changes) between the biodiversity system and the climate system. Interactions are studied at various scales, ranging from microscopic to macroscopic


CF'15, May 18 - 21, 2015, Ischia, Italy.
Copyright 2015 ACM 978-1-4503-3358-0/15/05...$15.00
http://dx.doi.org/10.1145/2742854.2747282
.

scales, and on (genomic, taxonomic, ecosystem) scales of level of individual plant and animal species. Methodologies with analyses using integrated indicators are more informative than only simple direct measurements of a few parameters. In this use case, the monitoring changes in biodiversity over a landscape is addressed through the temporal analysis of selected indicators (change analysis) and, for example, by using Ecological Niche Modelling (ENM) [3] techniques to understand changes in species geographic distribution within future scenarios.
Workflows have to be produced to combine the analysis of data acquired with different technologies, such as LiDAR, hyper-spectral imagery, satellite images and ground level sensors, with meteorological and biodiversity data to study the impact of climate change in regions with high interest for biodiversity conservation, such as the Brazilian Amazon and the semi-arid Caatinga regions in Brazil. The analysis of remote sensing images provides 3D information concerning the structure of the vegetation, such as the biomass distribution within the forest canopy and forest gap density patterns, which improves biodiversity indicators such as the energy balance and evapotranspiration.
To address all these scientific challenges, the use case joins together heterogeneous data sources, on-premises cloud infrastructures, multiple data services, and a scientific gateway into a single, federated environment.
The co-operation from Brazilian and European centres is key, as expertise in biodiversity modelling and related data come from the Brazilian side, while the expertise on climate change data analysis, together with the access to the ESGF [4] federated data archive and additional remote sensing data sources, come from both the European and Brazilian side.
In the remainder of this work, the architectural design of the use case environment, as well as its infrastructural implementation, are presented and discussed in detail. A special focus is devoted to the cloud-based parallel data analytics service exploited in the project to analyse large volumes of climate, satellite, and LiDAR datasets.

# 3. USE CASE ARCHITECTURE

The architecture of the use case consists of the following six components: user application, infrastructure management, federation layer, infrastructure resources, analytics as a service, and security.
The *user application* consists of two different components: the front-end and the back-end. The former provides the application front-end and the business logic related to the presentation layer, to manage the users requests. The latter is related to additional components able to translate the users requests into data-driven workflows. Such components include smart job scheduling over dynamically and elastically deployed set of data analytics services in the cloud. The *infrastructure manager* is responsible for the provisioning of resources. It interacts with the *federation layer*, which is responsible for federating cloud resources on top of the IaaS layer. The *infrastructure resources* consist of storage, network and computational resources. Data sources are also available from different data providers and are part of the use case too. *Security* is orthogonal in the architecture, which means it must be addressed vertically at each layer, from the user application to the infrastructure resources. Finally, the *analytics as a service* is a specific PaaS layer providing the framework needed to run the use case analytics tasks in the cloud.

# 4. USE CASE INFRASTRUCTURE

The system infrastructure (see Fig. 1) includes several components and services. Next we follow a bottom-up approach to present a complete description of each of them, their interactions, interfaces and features.

## 4.1 Infrastructure layer

The infrastructure layer exploited in this use case consists of several private clouds running OpenNebula or OpenStack at the Infrastructure as a Service (IaaS) level.
The data sources made available by the project partners, or already available from national and international agencies, are part of the infrastructure too, with a more static setup. In particular, the main data sources selected for this use case are:

- Climate data from the CMIP5 Federated Data Archive (ESGF) [5]. The Coupled Model Intercomparison Project provides a community-based infrastructure in support of climate model diagnosis, validation, intercomparison, documentation and data access. CMCC provides about 100TB of data related to three different models, NetCDF format [6], CF conventions. Starting from these datasets, multiple climate indicators can be computed (e.g. TXx, monthly maximum value of daily maximum temperature);

- LiDAR data. For the areas near Manaus in Brazil, where hyper-spectral imagery is apparently absent, LiDAR data are provided by EMBRAPA [7], Brazilian Agricultural Research Corporation. They represent the most important data sources for the extraction of 3D vegetation information. Vegetation and terrain metrics represent the key indicators that can be inferred from these datasets.

- SEBAL datasets. They are an output of satellite images series (LANDSAT and MODIS) processed by the SEBAL [8], [9] algorithm (which is out of the scope of this paper) to produce estimates of energy balance and evapotranspiration of water to the atmosphere (e.g. Enhanced Vegetation Index (EVI), Normalized Difference Vegetation Index (NDVI), Leaf Area Index (LAI), Surface Temperature (Ts), Albedo). Remote sensing data are provided by the United States Geological Survey (USGS) and the National Aeronautics and Space Administration (NASA).

- Species data [10]. The speciesLink datasets are an output of networking activities to provide free and open access to 7.3 million primary research-grade data, derived from the federation of 350 Brazilian biodiversity datasets, gathered from 150 institutions in Brazil and abroad. They represent valuable biodiversity data sources. In this context, species occurrences provide an indication about the presence of some species in a specific area.

## 4.2 Federation layer

In the EUBrazilCC project the fogbow middleware [11] is being exploited to implement the federation layer. It

addresses key federation-level challenges like the management of federation membership, the match-making policy for requesting and providing resources, the distributed authentication and authorisation of both federation and local users, as well as automatic setup of tunnels to allow virtual machines with private IPs to be accessed from outside the private clouds where they run.

There is a fogbow manager component running on top of each local private IaaS that wants to join a fogbow federation. Fogbow uses designed-to-federate and internet-friendly technologies such as XMPP, and it is flexible enough to deal with a wide range of cloud technologies. This is achieved thanks to a plugin framework that facilitates the provision of specific implementation for authentication (for users and members), authorization, image storage management and compute services. It is important to mention that fogbow provides a standard OCCI [12] interface implementation through which clients can interact with a particular fogbow manager.

## 4.3 Security model

The security model adopted in the federated cloud works in two levels: federation and local. An entity — user or program — interacting with a fogbow manager must provide its federation credential and, optionally, its local credential, valid in the underlying cloud associated to the fogbow manager contacted. These credentials are used to define not only who is entitled to instantiate VMs in the federated cloud, but also the quota of resources that is associated to different entities. There is a quota established for the use of the federation at each member private cloud. By providing its local credential, an entity is able to access extra resources using its quota in the underlying local cloud. In this use case we use a Virtual Organisation Membership Service (VOMS) [13] to provide X509v3 proxy digital certificates as credentials at the federation level, and the native identity service of the underlying cloud to provide local credentials. When the native identity service supports VOMS credentials, then the federation credential can also be used as a local credential. We note that this security model is compatible with that of the EGI Federated Cloud [14].

## 4.4 Infrastructure Manager

The Infrastructure Manager (IM) is a tool that eases the access and the usability of IaaS clouds by automating the VMI selection, deployment, configuration, software installation, monitoring and update of Virtual Appliances [15]. Such a service provides its functionalities through a set of APIs (XML-RPC and REST APIs). IM uses the Resource and Application Description Language (RADL) language to create and to get information about the infrastructure. RADL is a custom language that allows specifying the requirements of the resources where the scientific applications will be executed. Besides hardware specification (CPU, RAM, network), it also addresses software requirements (applications, software libraries, database systems) and the configuration of operating systems and applications. It uses a declarative approach that merges standard specifications, such as OVF [16] with the contextualization language derived from Ansible [17]. The current testbed exploits IM to instantiate Virtual Machine Images (VMI) on the private cloud infrastructure.

The VMIs are listed in the Virtual Machine Image Repository and Catalogue (VMRC) [18], which is used by the IM

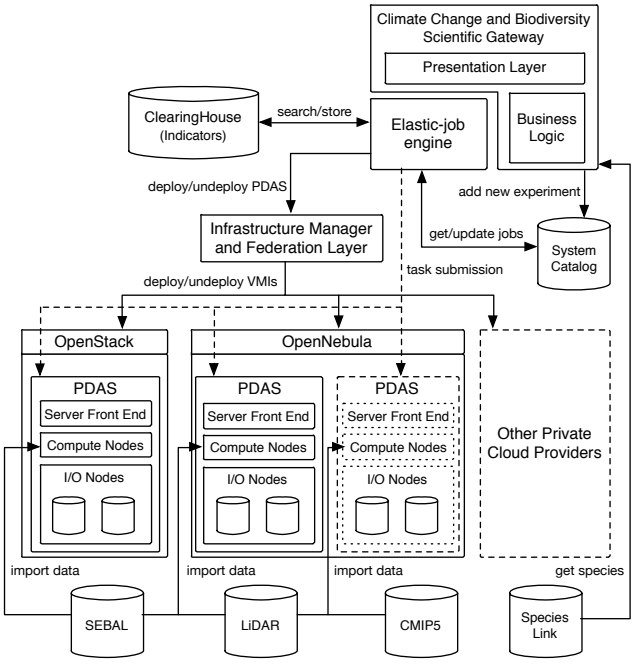


**Figure 1: System infrastructure**

to find a suitable VMI that accomplishes the requirements of the user (in terms of operating system, CPU architecture, installed applications, etc.), and it is compatible with the hypervisor available in the cloud system. This component indexes VMIs in order to be reused in multiple contexts. It also implements matchmaking algorithms to obtain a ranked list of VMIs that satisfy the requirements. IM uses fogbow OCCI API to issue the requests for creating VMs in the federated infrastructure.

## 4.5 Parallel Data Analytics Service

The Parallel Data Analytics Service (PDAS), a core component of the Ophidia project [19], [20], [21], addresses big data analytics challenges in multiple scientific domains. It executes data-intensive analysis on multi-terabytes datasets, exploiting advanced parallel computing techniques and smart data distribution methods. It provides a framework for parallel I/O and data analysis, an array-based storage model and a hierarchical storage organisation to partition and distribute multidimensional scientific datasets.

### *4.5.1 PDAS architecture*

From an architectural point of view, the PDAS consists of the following main layers: PDAS front-end, compute nodes, I/O nodes and storage.

The front-end provides a new GSI/VOMS enabled interface jointly with the initial WS-I-based. The grid-enabled interface addresses the interoperability requirement and the security model defined in the EUBrazilCC project. Regarding the authorisation, the PDAS front-end supports three different modes: (i) local, based on Access Control List - ACL, (ii) global, based on VOMS attributes and (iii) combined, which basically exploits the two previous modes joining fine tuning of the first mode with the flexibility and the scalability of the second one.

Concerning the compute nodes, they are used by the PDAS to run parallel (MPI-based [22], [23]) tasks, named *operators*, on the multidimensional data managed by the framework.
On the other hand, the I/O nodes are essentially devoted to the parallel I/O and analytics. It is worth mentioning each I/O node consists of one or more I/O servers responsible for the I/O with the underlying storage system, which in turn provides the hardware resources to manage the entire data store.
The internal storage model is based on a key-value pair approach to store scientific multidimensional data and provide the data cube abstraction. From a physical point of view, a data cube is horizontally partitioned into several blocks (called fragments) that are distributed across a set of I/O nodes. The production-level release of the PDAS exploits MySQL servers at the parallel I/O level. The new one (still in alpha release) provides a native in-memory I/O server with real-time analytics capabilities.
The I/O servers support the management and analysis of $n$-dimensional arrays (e.g. timeseries) through an extensive set of array-based primitives. To address flexibility, the primitives are designed and implemented as dynamic libraries to be plugged in different I/O servers without any effort. Furthermore, plugins can also be nested into each other to enable more complex tasks.
The currently available array-based primitives allow data sub-setting, data aggregation, array concatenation, algebraic expressions, predicate evaluation and compression routines. Core functions of well-known numerical libraries (e.g. GSL [24], PETSc [25]) and tools have been included into specific primitives.
According to the data cube abstraction, the PDAS provides many parallel operators to manipulate multidimensional datasets. Some examples are: datacube slicing, dicing, intercomparison, aggregation, array-based analysis, import, export.
Additional details about the PDAS (internal storage model, benchmark results, massive data experiments, operators, etc.) are available in previous works [19], [20], [21].

### 4.5.2 The PDAS role in the EUBrazilCC project

In the context of the EUBrazilCC project, the PDAS aims at elastically and dynamically addressing data analytics requirements concerning climate change and biodiversity in a federated cloud environment. The PDAS is being exploited to support the execution of data-driven workflows for the analysis of climate and biodiversity indicators.
Due to the framework peculiarity to deal with the multidimensional *data cube* abstraction, the PDAS has been extended to support the import of Landsat datasets (GeoTIFF format) and LiDAR data products (the support for the NetCDF format was already available).

### 4.5.3 Three cloud-based deployment scenarios

The PDAS can be deployed in different cloud-based scenarios according to scalability, performance and ease of deployment requirements. In this section, three general scenarios that could satisfy the majority of PDAS use cases are presented and discussed (see Fig. 2).
In the first scenario (Deployment A) a single virtual machine containing all the PDAS components is deployed. This solution satisfies use case scenarios related to a very simple and

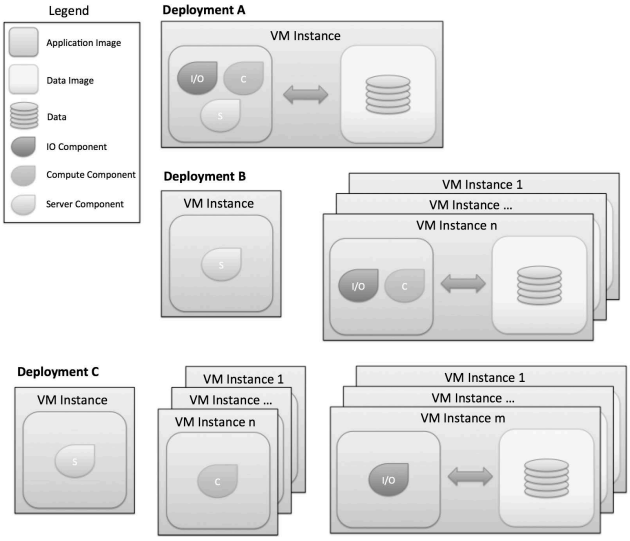


**Figure 2: Cloud-based deployment scenarios for the parallel data analytics service**

easy platform where the compute and I/O workload are low.
The second scenario (Deployment B) provides a good trade-off for scalability, performance and deployment simplicity; the scalability is achieved by using multiple instances of compute/IO components. In this scenario, the PDAS will be deployed by considering 2 VMIs: one with the server component and one with the compute and IO/data components. This scenario is particularly suitable for compute intensive data analysis jobs/workflows that require parallel execution by considering multiple instances of the compute and IO/data VM images.
The most flexible scalability and performance goals could be reached by considering a third scenario (Deployment C), where the PDAS could be deployed by considering 3 different VM images: one with the server, one with the compute, and finally one with the IO/data components. According to the compute and data workload, the scalability and high performance goals are achieved by instantiating several virtual machines related to the compute and IO/data images. This scenario is especially suitable for jobs that analyse large amounts of scientific data. It is worth mentioning that this scenario could include mounting the data repository from an external storage (e.g. via NFS) to share it across multiple PDAS cluster instances.
All of the three scenarios have been implemented, tested and validated in the private cloud environment available at CMCC using IM and fogbow. In this regard, Ansible scripts have been provided for the contextualisation of a PDAS cluster, which means, at lower level and in terms of VMIs, the configuration of a PDAS server front-end, a set of compute nodes and another set of I/O nodes. The VMIs corresponding to each scenario have been stored in the VMRC repository setup in the project.

### 4.5.4 A RADL request to setup a PDAS cluster

In our use case, to deploy a PDAS cluster in a private cloud, a RADL request must be submitted to IM. In the following, the most relevant sections of the RADL request used in the project for a PDAS cluster setup are presented.

First of all, two networks must be defined for a PDAS cluster: one public for the server front-end and another one, private, used by the server front-end, the I/O and compute nodes for the intra-cluster communications.

*network public_net (outbound='yes')*
*network private_net (outbound='no')*

In the following, we present the PDAS server specification. As it can be seen, the script includes the server specification in terms of cores, architecture, memory, OS and networks (both the public and the private one).

*system oph_server (*
*cpu.arch='x86_64' and*
*cpu.count=1 and*
*memory.size=1g and*
*net_interface.0.connection = 'public_net' and*
*net_interface.1.connection = 'private_net' and*
*disk.0.image.url='one://<ip>:2633/5' and*
*disk.0.os.name = 'linux' and*
*disk.0.applications contains (name='<ophidia-ansible-role>')*
*and*
*disk.0.os.credentials.username = '<user>' and*
*disk.0.os.credentials.password = '<passwd>'*
*)*

A similar specification is defined for the compute and I/O nodes. In this case, the only difference is related to the network, which is only private for the two components (here is reported the compute section only).

*system oph_computes (*
*...*
*net_interface.0.connection = 'private_net' and*
*...*
*)*

In the following, the RADL file provides the roles for the PDAS server, I/O and compute nodes (here is reported the role for the PDAS server only, as the ones for the I/O and compute nodes are almost identical):

*configure oph_server (*
*begin*
*- roles:*
*- { role: '<oph-role>', oph-repo: '<url>', oph_node_type: 'server' }*
*end*
*)*

Here is the contextualization section for the PDAS cluster:

*contextualize (*
*system oph_server configure oph_server step 1*
*system oph_computes configure oph_computes step 2*
*system oph_ios configure oph_ios step 2*
*)*

Finally, the high-level PDAS cluster deploy request:

*deploy oph_server 1*
*deploy oph_computes 2*
*deploy oph_ios 4*

As it can be seen, this section specifies the cluster configuration as composed of a single server, two compute nodes and four I/O nodes.

### 4.5.5 Cloud-based extensions of the PDAS Terminal

To be able to manage in a simple and effective manner the capabilities made available by a PDAS cluster, a robust, comprehensive, effective and extremely usable client has been developed. It is a real terminal (or *shell*) with useful features such as the management of the commands history, the management of specific environment variables, a manual with the description of commands and variables, the management of key combinations for the smart-editing of the command line and more.
Moreover, it (i) supports user authentication - SSL or X509v3 certificates based - (ii) provides the proper environment to submit data cube operators to a PDAS server, as well as to retrieve and display the related results, and (iii) embeds a useful and complete user manual describing in detail operators goal, parameters, default values, etc.
For the use case purposes, the PDAS terminal has been extended with four additional commands that relate to the PDAS-IM interaction:

- *oph-deploy* <header_file> <radl_file> <IM_url>
- *oph-deploy_status* <header_file> <IM_url>
- *oph-get_server* <header_file> <IM_url>
- *oph-undeploy* <header_file> <IM_url>

where the *header_file* can include optional settings, the *radl_file* refers to the RADL request, and *IM_url* relates to the IM endpoint.
This means the PDAS terminal offers the opportunity to deploy a PDAS cluster in the cloud (submitting a request to IM), run a complete user session and perform a final undeploy of the previously allocated resources.

### 4.5.6 Analytics workflows for indicators

The PDAS cluster instances are exploited in this use case to compute climate and biodiversity indicators. A typical experiment consists of multiple indicators over a specific area and a well-defined data analytics workflow is associated to each indicator. The scientific workflows needed in our use case are Direct Acyclic Graphs (DAGs) of operators. They have been defined jointly with the application-domain scientists by exploiting a Data Analytics Workflow Modelling Language (DAWML, developed in the context of this project), a high-level language providing a complete set of components representing all of the needed data cube operators. Each component in the DAWML (i) is visually represented by a circle, (ii) is associated to a specific operator, (iii) expresses the dimensionality of the output, (iv) provides (when available) additional metadata information useful to better understand how the operator has to be applied.

With regard to the climate datasets, the *monthly maximum value of daily maximum temperature (TXx)* represents an interesting example of climate indicator. The related workflow (see Fig. 3) is represented by a pipeline consisting of the following data cube operators: (i) import of the input dataset, (ii) spatial sub-setting over the selected area,

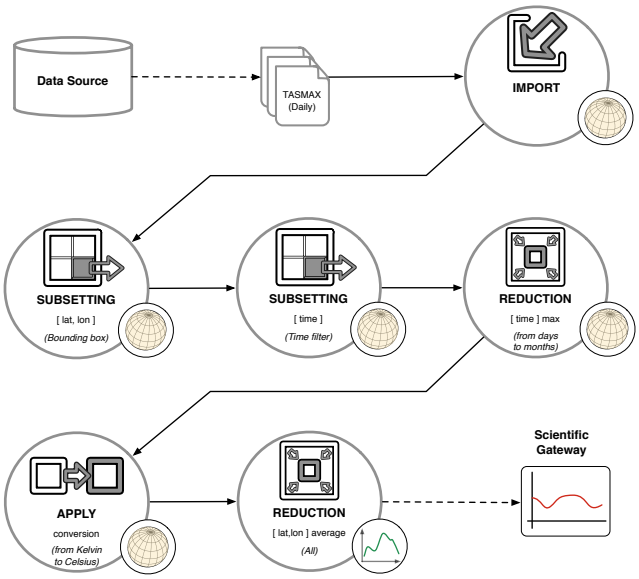


**Figure 3: A workflow related to the monthly maximum value of daily maximum temperature indicator (TXx)**

(iii) temporal sub-setting over the selected time domain, (iv) max reduction (from days to months), (v) unit conversion (from Kelvin to Celsius degrees), (vi) spatial reduction (mean over lat-lon). The output of this workflow is a time series (1D data cube in the PDAS).
Additional workflows have been also defined for the other indicators, namely vegetation indices and vegetation and terrain metrics defined in the use case and related respectively to the SEBAL and LiDAR data. With regard to the biodiversity data, analytics workflows involving the PDAS are not needed, as the only indicator available, so far, is related to the species occurrences which can be computed by querying the SpeciesLink service available at CRIA [26].

## 4.6 User application back-end

In the use case infrastructure, the *elastic-job engine* and the *ClearingHouse system* represent two key components of the user application back-end.
The elastic-job engine is responsible for providing dynamic job scheduling on the federated infrastructure (by interacting with already available PDAS cluster instances) and elastic deploy of PDAS cluster instances in the cloud (by interacting with IM). A new PDAS instance is automatically deployed in the cloud, when the set of pending tasks on the running instances exceeds a defined (configurable) threshold. This allows to horizontally scale in terms of PDAS clusters, according to the number of users requests.

The elastic-job engine schedules the tasks on the entire set of available PDAS instances according to the following policy: *each task is allocated to the instance with the lowest number of pending tasks over the entire set of available resources, until the threshold is reached.*
This approach minimises the overall makespan (considering constant the available set of resources). A new PDAS cluster is instantiated when the queues of all the active clusters have achieved the maximum threshold.
To understand the rationale behind this approach, we have

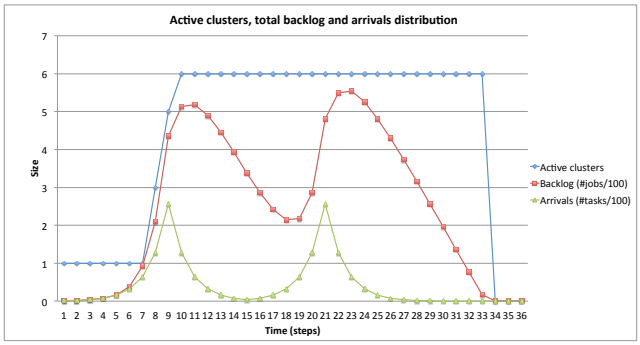


**Figure 4: Number of active clusters, total backlog and arrivals distribution in a two-burst scenario**

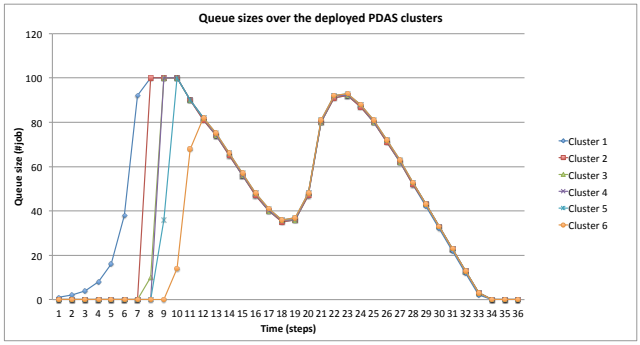


**Figure 5: Queue sizes over the entire set of PDAS cluster instances in a two-burst scenario**

implemented a simple simulator in C language, configured with a distribution related to the incoming requests (even including bursty behaviours) and the threshold for the queue to deploy a new PDAS cluster. As a general assumption, the tasks have been considered homogeneous. Fig. 4 shows the arrivals distribution of a simulation with two bursts (over 34 time steps), the number of PDAS clusters dynamically instantiated to serve the requests, and the total backlog across the entire set of active PDAS clusters. As it can be seen, after the first burst, the total number of active clusters is six and it is able to manage the remaining set of requests (even including another burst) without deploying additional resources. In such a phase the queues manage the same number of requests due to the adopted policy. Fig. 5 shows the queue sizes for the different PDAS clusters and shows how the requests are distributed in a way which balances the backlog associated to each queue. The simulation relates to a threshold for the task queues set to 100 tasks.
A more extensive analysis related to the results of these simulations and the associated mathematical model, cannot be presented in this paper due to space limitations. However, general results from our simulations can be summarised as it follows: the higher is the threshold, the lower is the number of deployed clusters, as more requests can be allocated on the already active PDAS instances. This, of course, negatively affects the backlog and the overall makespan, which increases.
On the other hand, the lower is the threshold, the higher is the number of deployed clusters, especially close to the bursts of requests. In this case a higher overhead related to

the deploy (and undeploy) of new cluster instances has to be considered.
Another component in the back-end of the user application is the ClearingHouse system, a persistent database where users can store/publish the results of their experiments. Such a database represents an interesting knowledge base available to all the users. Statistics on these experiments are also tracked in order to highlight in the gateway those ones that have higher impact on the end-users (top indicators list).

## 4.7 User application front-end

In terms of front-end, the *Climate and Biodiversity Scientific Gateway* provides the proper access point to scientists for running their experiments. The gateway has been designed and implemented according to the Model-View-Controller (MVC) pattern. In terms of requirements, it has to provide:

- *experiment definition*: through a high-level web interface, the user should define her own experiment by selecting a set of climate and biodiversity indicators as well as the spatial, through GoogleMaps, and temporal domains for the experiment (see Fig.6). For climate indicators, future emissions scenarios like RCP8.5 and RCP4.5 [27] can be considered too;
- *submission and monitoring of the analytics tasks*: a user-friendly interface allows the submission and dynamic monitoring of the experiments (and related workflows) status (see Fig.6). Failures are also reported and re-submit policies are available too;
- *output visualization*: the user can get access to the output of the experiment (e.g. timeseries related to a set of indicators), display them (e.g. as line charts or histograms), perform visual comparison of different indicators as well as download/export of the output results (see Fig.7);
- *clearinghouse management*: the user can persistently store into a database the output of a specific experiment, as well as perform search & discovery tasks to get fast access to the output of the already available experiments (for instance, performed by other users) avoiding to run the same experiment on the infrastructure several times.

The Scientific gateway communicates with the elastic-job engine through the system catalogue (a system database), which stores the user's requests, the set of experiments, their status, as well as additional metadata.
Finally, from a technical standpoint, the gateway is a web application exploiting the Javascript language and the ExtJS framework for the front-end jointly with a lightweight set of Java actions for the business logic.

# 5. MAJOR IMPACT AND ADDED VALUE

The *major impact* on the end-user community is related to the integrated environment. Indeed, the possibility to perform data integration, processing, analysis and visualisation across multiple, heterogeneous data sources into one, integrated platform, is a major step forward for the scientific discovery in this research context.

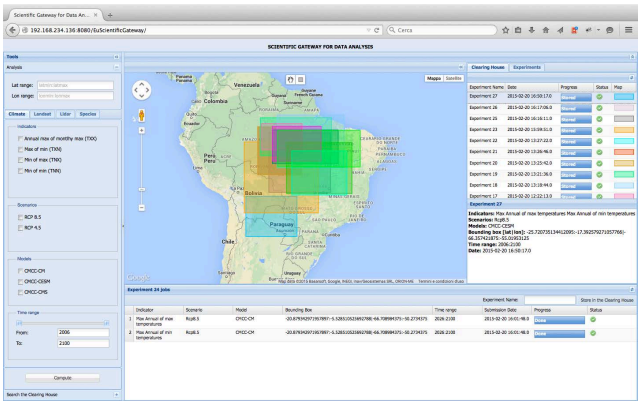

**Figure 6: Visualisation of multiple indicators in the Scientific gateway**

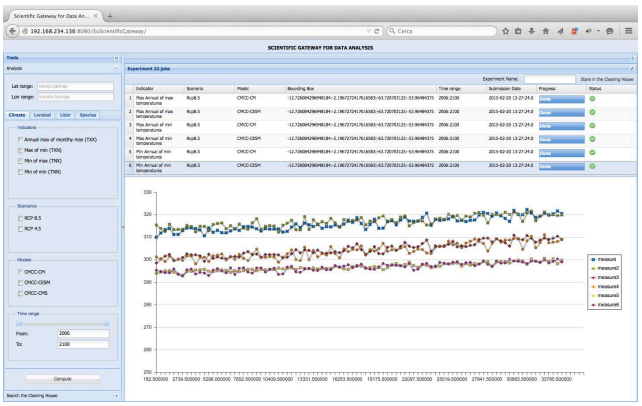

**Figure 7: Visualisation of multiple indicators in the Scientific gateway**

The *added value* comes from the following factors provided in the proposed environment: (i) various data types and products are brought together in the infrastructure; (ii) the data comes in a variety of formats, scales and extents, which are all aligned in the infrastructure for processing; (iii) data sources are analysed along transparent workflows and processing pipelines; (iv) the infrastructure allows processing newly available datasets; (v) new analytics clusters can be dynamically and elastically allocated according to the jobs load (which was not the case in previous static cluster-based deployments); (vi) end-users are provided with a platform that allows quick retrieval of the products stored in the infrastructure through a clearinghouse-based approach; (vii) on-demand processing of selected datasets allows a quick review of the resulting information, which increases understanding and discussion amongst researchers from different scientific disciplines; finally, (viii) a strong support for data-driven workflows in the climate and biodiversity domains.

# 6. CONCLUSIONS AND FUTURE WORK

This work presents the use case environment implemented during the first year of the EUBrazilCC project. The most relevant developments are related to the implementation of the elastic-job engine, the scientific gateway and the PDAS extensions to run, as a PaaS, in a private cloud environ-

ment. The different components of the infrastructure are being tested and validated with regard to two areas in Brazil (Caatinga and Manaus). However, as pointed out by the application-domain scientists involved in the project, the scientific validation of data processing pipelines in small areas, like the two ones selected in the context of the project, paves the way for computation of larger areas in the cloud infrastructure, which was so far difficult, or impossible using traditional computers.

The future work mainly concerns the activity foreseen in the second year of the project, which means (i) the integration of correlation analysis regarding the different indicators, (ii) a complete integration of the federation layer to include additional private cloud providers, (iii) new PDAS operators and primitives to completely address the use case analytics requirements and (iv) the integration of new analysis and visualisation functionalities in the scientific gateway.

## 7. ACKNOWLEDGMENTS

This work was supported by the EU FP7 EUBrazilCC Project (Grant Agreement 614048), and CNPq/Brazil (Grant Agreement 490115/2013-6).

## 8. ADDITIONAL AUTHORS

Additional authors:
Iana A. A. Rufino, UFCG, iana.alexandra@ufcg.edu.br,
Arie C. Seijmonsbergen, IBED, email: a.c.seijmonsbergen@uva.nl,
Carlos de Oliveira Galvao, UFCG, email: galvao@dec.ufcg.edu.br,
Vanderlei Perez Canhos, CRIA, email: vcanhos@cria.org.br,
Andrea Mariello, CMCC, email: andrea.mariello@cmcc.it,
Cosimo Palazzo, CMCC, email: cosimo.palazzo@cmcc.it,
Alessandra Nuzzo, CMCC, email: alessandra.nuzzo@cmcc.it,
Alessandro D'Anca, CMCC, email: alessandro.danca@cmcc.it,
Giovanni Aloisio, CMCC, email: giovanni.aloisio@cmcc.it

## 9. REFERENCES

[1] S. Fiore and G. Aloisio, 2011. Special section: Data management for eScience. *FGCS* 27(3), 290-291.

[2] EUBrazilCC http://eubrazilcloudconnect.eu

[3] M.E.S. Munoz, R. Giovanni, M.F. Siqueira, T. Sutton, P. Brewer, R.S. Pereira, D.A.L. Canhos, & V.P. Canhos, (2009) openModeller: a generic approach to species potential distribution modelling. *GeoInformatica*, DOI: 10.1007/s10707-009-0090-7

[4] L. Cinquini, D. Crichton, C. Mattmann, J. Harney, G. Shipman, F. Wang, R. Ananthakrishnan, N. Miller, S. Denvil, M. Morgan, Z. Pobre, G. M. Bell, C. Doutriaux, R. Drach, D. Williams, P. Kershaw, S. Pascoe, E. Gonzalez, S. Fiore, R. Schweitzer, The Earth System Grid Federation: An open infrastructure for access to distributed geospatial data, *FGCS*, ISSN 0167-739X, http://dx.doi.org/10.1016/j.future.2013.07.002.

[5] K. E. Taylor, R. J. Stouffer, and G. A. Meehl, 2012, An overview of CMIP5 and the experiment design. *Bulletin of the American Meteorological Society* 93(4), 485-498, http://dx.doi.org/10.1175/BAMS-D-11-00094.1.

[6] R. K. Rew and G. P. Davis, 1990, The Unidata NetCDF: Software for scientific data access. In *6th Int. Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*, American Meteorology Society, 33-40 (February 1990).

[7] EMBRAPA https://www.embrapa.br/

[8] Bastiaanssen, W. G. M., Menenti, M., Feddes, R. A., et al. A remote sensing surface energy balance algorithm for land (SEBAL): 1. Formulation. Journal of Hydrology, v. 212 - 213, p. 198 - 212, 1998a.

[9] Bastiaanssen, W. G. M., Pelgrum, H., Wang, J., et al. A remote sensing surface energy balance algorithm for land (SEBAL): 2. Validation. Journal of Hydrology, v. 212 - 213, p. 213 - 229, 1998b.

[10] speciesLink. Centro de Referencia em Informacao Ambiental, Campinas. http://splink.cria.org.br/.

[11] fogbow http://www.fogbowcloud.org/

[12] OCCI http://occi-wg.org/

[13] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, A. Frohner, K. Lorentey, and F. Spataro, From gridmapflle to voms: managing authorization in a grid environment. *FGCS*, 21(4):549-558, 2005.

[14] EGI FedCloud http://www.egi.eu/infrastructure/cloud/

[15] M. Caballer, I. Blanquer, G. Molto and C. de Alfonso, Dynamic management of virtual infrastructures. *Journal of Grid Computing*, 2014, ISSN 1570-7873, 10.1007/s10723-014-9296-5.

[16] DMTF Open Virtualization Format Specification version 1.0.0, DSP0243, 22 February 2009.

[17] Ansible http://www.ansible.com.

[18] J. V. Carrion, G. Molto, C. De Alfonso, M. Caballer, V. Hernandez, A Generic Catalog and Repository Service for Virtual Machine Images, in: *2nd International ICST Conference on Cloud Computing* (CloudComp 2010), (2010).

[19] S. Fiore, A. D'Anca, C. Palazzo, I. Foster, D. Williams and G. Aloisio, 2013, Ophidia: Toward Big Data Analytics for eScience. *ICCS 2013*, 2376-2385.

[20] S. Fiore, C. Palazzo, A. D'Anca, I. T. Foster, D. N. Williams, G. Aloisio: A big data analytics framework for scientific data management. *IEEE BigData Conference* 2013: 1-8

[21] S. Fiore, A. D'Anca, D. Elia, C. Palazzo, I. Foster, D. Williams, G. Aloisio, Ophidia: A Full Software Stack for Scientific Data Analytics, proc. of the *2014 International Conference on High Performance Computing & Simulation* (HPCS 2014), July 21 - 25, 2014, Bologna, Italy, pp. 343-350, ISBN: 978-1-4799-5311-0.

[22] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, MIT Press, 1999.

[23] W. Gropp, E. Lusk, and R. Thakur, *Using MPI-2: Advanced Features of the Message-Passing Interface*, MIT Press, 1999.

[24] The GNU Scientific Library (GSL) http://www.gnu.org/software/gsl/.

[25] Satish Balay, Jed Brown, Kris Buschelman, William D. Gropp, et al., PETSc web page http://www.mcs.anl.gov/petsc, 2012.

[26] SpeciesLink service http://tapir.cria.org.br/tapirlink/tapir.php/specieslink

[27] Emission scenarios http://www.wmo.int/pages/themes/climate/emission_scenarios.php