Procedia Computer Science

Volume 80, 2016, Pages 722–733

ICCS 2016. The International Conference on Computational Science

Two-level dynamic workflow orchestration in the INDIGO DataCloud for large-scale, climate change data analytics experiments

Marcin Płóciennik[1], Sandro Fiore[2], Giacinto Donvito[3], Michał Owsiak[1], Marco Fargetta[3], Roberto Barbera[3], Riccardo Bruno[3], Emidio Giorgio[3], Dean N. Williams[4], and Giovanni Aloisio[2]

[1] Poznan Supercomputing and Networking Center IBCh PAS, Poznan, Poland
marcinp@man.poznan.pl,michalo@man.poznan.pl
[2] Fondazione Centro Euro-Mediterraneo sui Cambiamenti Climatici, Lecce, Italy
sandro.fiore@cmcc.it,giovanni.aloisio@cmcc.it
[3] Istituto Nazionale Fisica Nucleare
giacinto.donvito@ba.infn.it,marco.fargetta@ct.infn.it, roberto.barbera@ct.infn.it, riccardo.bruno@ct.infn.it, emidio.giorgio@ct.infn.it
[4] Lawrence Livermore National Laboratory, Livermore, USA
williams13@llnl.gov

Abstract

In this paper we present the approach proposed by EU H2020 INDIGO-DataCloud project to orchestrate dynamic workflows over a cloud environment. The main focus of the project is on the development of open source Platform as a Service solutions targeted at scientific communities, deployable on multiple hardware platforms, and provisioned over hybrid e-Infrastructures. The project is addressing many challenging gaps in current cloud solutions, responding to specific requirements coming from scientific communities including Life Sciences, Physical Sciences and Astronomy, Social Sciences and Humanities, and Environmental Sciences. We are presenting the ongoing work on implementing the whole software chain on the Infrastructure as a Service, PaaS and Software as a Service layers, focusing on the scenarios involving scientific workflows and big data analytics frameworks. INDIGO module for Kepler worflow system has been introduced along with the INDIGO underlying services exploited by the workflow components. A climate change data analytics experiment use case regarding the precipitation trend analysis on CMIP5 data is described, that makes use of Kepler and big data analytics services.

Keywords: Kepler, cloud, PaaS, Climate change, Ophidia, FutureGateway, INDIGO

 Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2016

doi:10.1016/j.procs.2016.05.359

1 Introduction

There are numerous areas of interest, for scientific communities, where cloud Computing utilization is currently lacking, especially at the PaaS (Platform as a Service) and SaaS (Software as a Service) levels. In this context, INDIGO-DataCloud [6] (INtegrating Distributed data Infrastructures for Global ExplOitation), a project funded under the Horizon 2020 framework program of the European Union, aims at developing a data and computing platform targeted at scientific communities, deployable on multiple hardware, and provisioned over hybrid e-Infrastructures. The project is built around the requirements coming from several research communities (Life Sciences, Physical Sciences and Astronomy, Social Sciences and Humanities, and Environmental Sciences) including the one representing the European Strategy Forum on Research Infrastructures (ESFRI) roadmap projects [22], like LifeWatch, EuroBioImaging, INSTRUCT, CTA, ELIXIR, EMSO, DARIAH. The core of the project activities is focusing on the development of an open source PaaS solution allowing public and private e-infrastructures, including those provided by EGI [3], EUDAT [4] and Helix Nebula [5], to integrate their existing services. In addition, the project aims to develop a flexible presentation layer connected to the underlying IaaS and PaaS frameworks. It will also provide the tools needed for the development of APIs to access the PaaS framework. Toolkits and libraries for different frameworks will be provided - including for scientific workflow systems like Kepler.

In the following sections the gaps of the currently available solutions, as well as the requirements coming from the scientific user communities are presented. Section 2 discusses the main challenges faced by the INDIGO project. Section 3 describes the architecture of the INDIGO DataCloud, highlighting the parts related to scientific workflows (and Kepler extensions), big data analytics, and scientific gateway. Section 4 presents the challenging big data analytics use case coming from Climate Change community, as well as architectural and infrastructural details. This use case integrates and uses most of the components being developed as part of INDIGO project. Finally, Section 5 draws the final conclusions and future work.

2 INDIGO challenges

To reach the full promises of cloud computing, major aspects have not yet been developed [20]. One of the main open issues is interoperation across (proprietary) cloud solutions. A second issue is dealing and assuring multi-tenancy in heterogeneous environments. Also dynamic and seamless elasticity from in-house cloud to public Clouds is not easy to be fulfilled. INDIGO-DataCloud is addressing those and a number of the other identified gaps including: (i) static allocation and partitioning of both storage and computing resources in data centers, (ii) current inflexible ways of distributing and deploying applications, (iii) lack of the dynamic workflow orchestration capabilities [14], and (iv) enhanced interfaces/APIs, also for tightly coupled big data analytics workflow support.

In the big data area, several frameworks (e.g. SciDB, Rasdaman or MapReduce-like implementations) address data analysis on large volumes of scientific data providing server-side capabilities, but with some differences in terms of support for parallelism, in-memory analysis, multi-dimensional storage models, etc. Such frameworks are limited in terms of tightly coupled dynamic workflow orchestration support in the cloud. Moreover, for several research communities like the climate change one, domain specific tools (e.g. like the Climate Data Operators (CDO [2]) or NetCDF Operators (NCO [28])) are mostly client-side, sequential and without workflow support/interfaces.

One of the common requirements, coming from the user communities involved in the project,

can be described by the following use case: a user community uses an application that can be accessed via GUI, but at the same time requires batch queue system as back-end. In addition to that it has unpredictable workload requirements and has well defined access profile for the user. The application consists of two main parts: the scientific Gateway (or workflow system) and the processing working nodes. These requirements imply that working nodes should scale-up and -down according to the workload. In particular (very demanding) cases the cloud-bursting to external infrastructures may be requested. In addition portal/workflow services should also adapt to workload. The whole list of the requirements is available in requirement analysis project deliverable [21].

3 INDIGO vision

3.1 General architecture

The INDIGO-DataCloud architecture consists of a number of technologies, components, and layers as presented in the Fig 1. This paper mainly focuses workflow aspects, and big data analytics for the presented large scale experiment. End users will be provided either with the Graphical User Interfaces (Scientific Gateways like FutureGateway, see Section 3.2, or Workflows Systems like Kepler, see Section 3.3) or simple APIs. Graphical User Interfaces will use the FutureGateway Engine and its JSAGA adaptors to access on the INDIGO PaaS Orchestrator services. The user authenticated on the INDIGO Platform will be able to access and customize a rich set of TOSCA-compliant templates, that is the language in which the INDIGO PaaS is going to receive the end-user request. TOSCA(OASIS Topology and Orchestration Specification for Cloud Applications) is an OASIS(Organization for the Advancement of Structured Information Standards) specification for the interoperable description of application and infrastructure cloud services, the relationships between parts of these services, and the operational behaviour of these services.

The PaaS Core provides an entry point to its functionality via the Orchestrator service. The PaaS core components will be deployed as a suite of small services using the concept of micro-service. Kubernetes, an open source platform to orchestrate and manage Docker containers, will be used to coordinate the micro-services in the PaaS. Orchestrator, among many other activities, will interact with the Application Deployment Service that is in charge of scheduling, spawning, executing and monitoring applications and services on a distributed infrastructure. The core of this component consists of an elastic Mesos [13] cluster with slave nodes dynamically provisioned and distributed on the IaaS sites. The Mesos cluster consists of one or more master nodes, and slave nodes that register with the master and offer resources. The Automatic Scaling Service, based on EC3/CLUES (Elastic Cloud Computing Cluster) [12], will ensure the elasticity and scalability of the Mesos cluster by monitoring its status. When additional computing resources (worker nodes) are needed, the Orchestrator will be requested to deploy them on the underlying IaaS (OpenStack, OpenNebula). Using the plugin architecture of Mesos, features like deployment of a batch cluster on demand will be developed. It will allow to run on demand solutions like Hadoop, or the front-end portals plus tightly coupled clusters.

3.2 FutureGateway services

FutureGateway is a framework developed for the scientific community. It is mainly based on web portals, defined as Science Gateway (SG) [27, 26], that provide access to remote e-Infrastructures. It is the result of constant evolution of the Catania Science Gateway Frame-

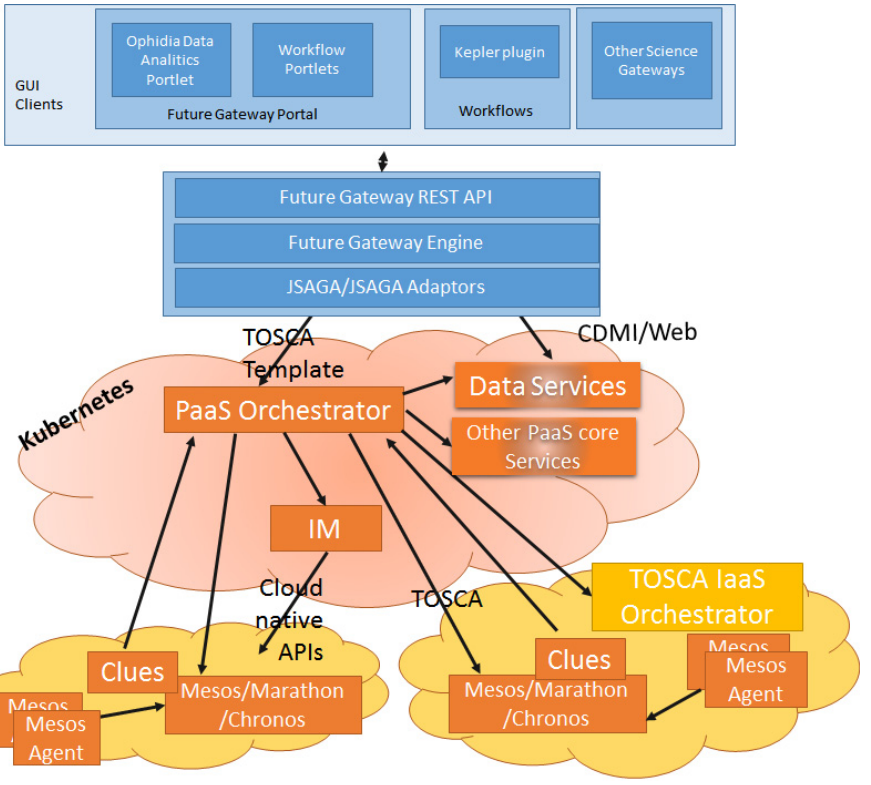

Figure 1: INDIGO-DataCloud overall architecture

work [10, 18] which has been re-engineered and extended to fulfil the requirements defined by the INDIGO-Datacloud use cases. It includes, but does not limit to: (i) more complex deployment scenarios, (ii) integration of web interfaces with desktop and mobile applications, and (iii) PaaS based e-Infrastructures developed in INDIGO-Datacloud. The main components of the FutureGateway framework are the Portal and the Engine. The former implements the web interface exposing the user applications to the community and hides all the interactions with the e-Infrastructures. The Portal is built on top of Liferay Application framework [8] and consists of a set of customisable portlets which can be integrated by the portal manager to create a Science Gateway. Additionally, the portal provides some common functionalities crosscutting all the activities in the SG such as the AutnN/AuthZ, task status management and others.

The FutureGateway Engine is the core component of the architecture. It consists of RESTful service intermediating between the e-Infrastructure (including PaaS Orchestrator) and high level services providing the graphical user interfaces. The RESTful APIs are designed to simplify development of portals, mobile and desktop applications. Internally the engine uses the JSAGA library [7] which is an implementation of the OGF SAGA standard [9]. The JSAGA implementation makes use of the adaptor design pattern to implement the connections with the e-Infrastructures. In this pattern, the high-level functions are associated with the adaptor at run time, according to the chosen infrastructure. This allows transparently executing the same applications on cloud, HPC and Grid resources. Additionally, a new set of adaptors is

under development in order to support the INDIGO-Datacloud use cases. These adaptors will implement the interaction with the INDIGO PaaS Orchestrator, based on TOSCA() templates, and CDMI data management.

3.3 INDIGO Kepler actors and module

The INDIGO module allows to utilize the underleying RESTful API exposed by FutureGateway Server. This way, an execution chain provided by the API can be easily formed as a workflow inside Kepler. Kepler provides default actor for RESTful client - *RESTService*. However, it has limitations that were preventing us from using it: a message is sent as attachment instead of being message's body, we have to use temporary files for JSON transfer, there is no implementation for DELETE method. Taking these limitations into consideration, we have decided to develop actors based on use cases rather than general RESTful client. Module delivers one actor per user's activity: prepare task, submit task, check task's status, upload data, remove task, etc. This approach was triggered by the fact that each function exposed by the API requires different inputs and can be executed with different HTTP request methods (PUT, GET, DELETE). Forcing users to pick proper set of parameters for each call would trigger lots of confusion.

The first version of the INDIGO module have been released but still gradually new functionalities are being added. At the moment, it is possible to build workflows that define task, prepares inputs and triggers execution. While a task is executed within INDIGO's infrastructure, it is possible to check its status, as presented in Fig. 2. Further developments will provide functionalities that will allow a complete management of infrastructure: (i) defining applications that can be executed, (ii) managing existing applications, (iii) task management (adding, removing, listing), and (iv) output handling. Eventually, the INDIGO module will provide the full stack of execution based on FutureGateway's APIs. The INDIGO module will extend current ways of distributed workflows execution, described in details in [16].

3.3.1 Workflow as a Service model for Kepler

Another model of running Kepler workflows is enabled by using the INDIGO PaaS and its feature - Automatic Scaling Service - based on EC3/CLUES. In this case the user will be able to instantiate on demand all the services required for the workflow execution, e.g. the workflow engine, the batch system and data/storage end points. The user will be provided with ready-to-use TOSCA recipes that can be customized using a friendly graphical interface, in order to adjust parameters like the size of the resources required for the workflow run, type of the cluster. The elasticity feature of the INDIGO PaaS will allow to increase/decrease the amount of required cloud resources. It will allow to execute on demand workflows with the Kepler on front-end and Hadoop cluster behind (e.g. in bioinformatics use cases). Another example are the complex physics workflows with parameter sweep sub-workflows. Thanks to this mechanism, they will be able to easily scale without needing static reservation of resources.

3.4 Ophidia analytics stack and INDIGO extensions

Ophidia is a big data analytics framework addressing data analysis challenges in several scientific domains. It provides parallel, server-side data analysis, an internal storage model to manage multidimensional datasets as well as a hierarchical data organisation to manage large volumes of scientific data. The framework includes a large set of array-based primitives (about 100) and parallel datacube operators (about 50) to process and analyze large volumes of multidimensional

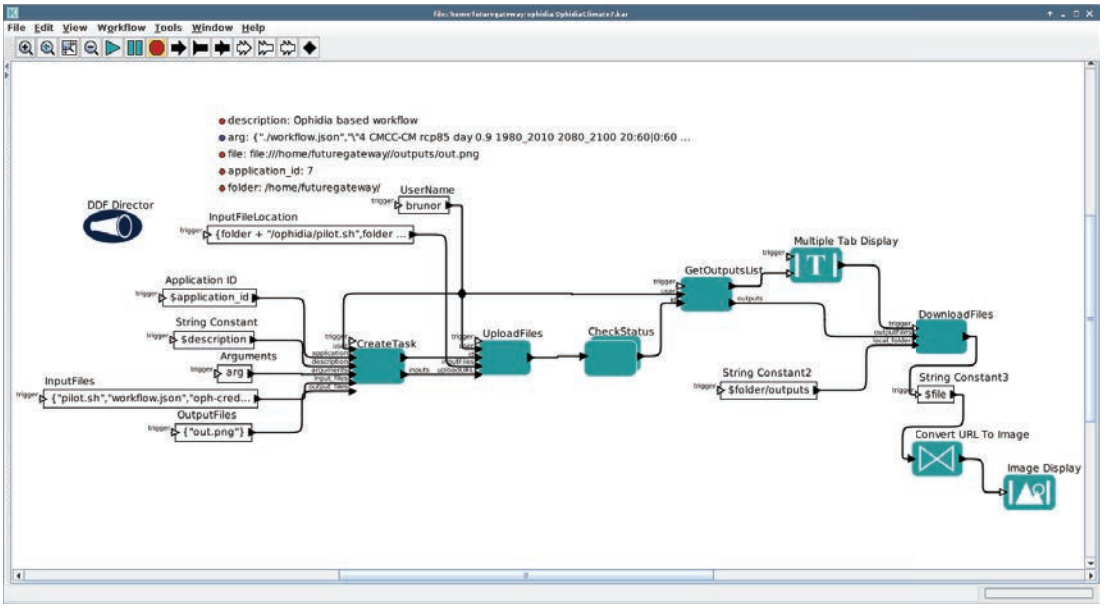

Figure 2: Example Kepler workflow using INDIGO actors

data. The Ophidia stack includes an internal workflow management system, which coordinates and orchestrates the execution of multiple scientific data analytics and visualization tasks (e.g. operational processing/analysis chains) at the server level. More details about the Ophidia architectural design and the infrastructural implementation can be found here [17, 19]. With regard to the state of the art, SciDB [25] and Rasdaman [11] are key projects that fall in the same research area of Ophidia, but with some key differences. While Ophidia implements a high performance OLAP approach leveraging on the datacube abstraction, SciDB and Rasdaman rely mainly on the array database concept (based on SQL at the low level). Some implementation differences are that SciDB relies on UDFs, Rasdaman on a dedicated array engine and Ophidia on a I/O in-memory engine able to run UDFs. An important feature related to the work presented in this paper and that is provided by Ophidia w.r.t. the other two systems, is the workflow support: Ophidia implements a native support for workflow management system [23] jointly with an internal workflow optimiser, which applies a set of equivalence rules (associated to the datacube algebra implemented in Ophidia) to increase the overall workflow performance. A detailed description of the datacube algebra is out of the scope of this paper and will be presented in a future work. To support the use case presented in this paper, the Ophidia workflow capabilities have been in particular extended in the INDIGO project to include:

- the massive interface, to apply the same task on multiple datasets with a single declarative statement. It is a pre-processing macro that filters out the input datasets from a large data collection, easing the definition of massive data analytics tasks. Although the massive filters can be applied to a diverse set of test cases, specific filters, based on the CMIP5 metadata, vocabulary and conventions have been implemented to further support scientific user scenarios in the climate change domain.

- the parallel interface, to apply the same set of tasks concurrently on different datasets.

Also in this case, it is a pre-processing macro providing inter-task parallelism support and specific filters to split computation on data across multiple parallel branches. It should be noted that, along with the workflow-based inter-task parallelism, each data operator in Ophidia is a MPI application (intra-task parallelism).

4 Climate Change use case

4.1 Earth System Models and the Climate Model Intercomparison Project (CMIP)

A major challenge for the climate change community is the development of comprehensive Earth system models capable of simulating natural climate variability and human-induced climate changes. Such models need to account for detailed processes occurring in the atmosphere, the ocean and on the continents including physical, chemical and biological processes on a variety of spatial and temporal scales. They have also to capture complex nonlinear interactions between the different components of the Earth system and assess how these interactions can be perturbed as a result of human activities. The development and use of realistic climate models requires a sophisticated software infrastructure and access to the most powerful supercomputers and data handling systems. In this regard, the increased models resolution is rapidly leading to very large climate simulations output that pose significant scientific data management challenges in terms of data processing, analysis, archiving, sharing, visualization, preservation, curation, and so on. In such a scientific context, large-scale experiments for climate model intercomparison (CMIP) are of great interest for the research community. CMIP provides a community-based infrastructure in support of climate model diagnosis, validation, intercomparison, documentation and data access. Large scale experiments like the CMIP*, have led to the development of the Earth System Grid Federation (ESGF [15]) a federated data infrastructure involving a large set of data providers/modeling centres around the globe. In the last 3 years, ESGF has been serving the Coupled Model Intercomparison Project Phase 5 (CMIP5) [1] experiment, providing access to 2.5PB of data for the IPCC AR5, based on consistent metadata catalogues. The ESGF infrastructure provides a strong support for: search and discovery, data browsing, data publication, data usage statistics, metadata management and (secured) access to climate simulation data and observational data products.

4.2 Current approach and limitations

The current scenario is based on a client side and sequential approach for climate data analysis consisting of the following steps: (i) search and discovery process across the ESGF federation; (ii) authentication via the ESGF Identity Provider and datasets download from the ESGF data nodes on the end-user local machine; and (iii) analysis steps. Datasets have to be downloaded from the ESGF data nodes on the end-users local machine before starting to run the analysis steps. The download phase in the second step is a strong barrier for climate scientists as, depending on the amount of data needed to run the analysis, it can take from days, to weeks, to months (i.e. ensemble analysis are often multi-terabyte experiments). Moreover, the intrinsic current nature of the approach also implies that end-users must have system management/ICT skills to install and update all the needed data analysis tools/libraries on their local machines. Another major critical point relates to the complexity of the data analysis process itself (third step). Analysing large datasets involves running multiple data operators, from widely adopted set of command line tools (e.g. CDO, NCO). This is usually done via scripts (e.g. bash) on

the client side and also requires climate scientists to take care of, implement and replicate workflow-like control logic aspects (which are error-prone too) in their scripts - along with the expected application-level part. The large amount of data and the strong I/O demand pose additional challenges to the third step related to performance. In this regard, production-level tools for climate data analysis are mostly sequential and there is a lack of big data analytics solutions implementing fine-grain data parallelism or adopting stronger parallel I/O strategies, data locality, processing chains optimization, etc.

4.3 Use case requirements

Starting from the issues described in the previous section, we present in the following, the key points resulting from the requirements analysis carried out in the first months of the project for the climate change research community:

- Efficiency/Scalability. Running massive inter-comparison data analysis involves large volume of scientific datasets (e.g. multi-terabyte order). There is a strong need to provide scalable solutions (e.g. HPC-, HTC-based) and more efficient paradigms (e.g. server-side) avoiding large data movement/download.

- Workflow support. Data analysis inter-comparison experiments are based on tens/hundreds of data analysis operators. Workflow tools could help managing the complexity of these experiments at different levels (multi-site and single-site) and increase the reusability of specific workflow templates in the community.

- Metadata management. It represents a complementary aspect that must be taken into consideration both from a technical (e.g. metadata tools) and a scientific (e.g. data semantics) point of view.

- Easy to use analytics environments. Providing an easy-to-use and integrated analytics environment could represent an added value to enable scientific research at such large scale.

- Interoperability/legacy systems. Interoperability with the existing ESGF infrastructure is key w.r.t. existing data repositories, interfaces, security infrastructure, data formats, standards, specifications, tools, etc.

From a technical point of view it also relates to having easy deployment procedures (e.g. cloud-based) to enable a larger adoption by the community.

4.4 Application of the INDIGO solutions

4.4.1 INDIGO approach and ambition

With regard to the current user scenario based on the three-step simple workflow mentioned in Section 4.2, INDIGO aims at providing (through the mapping of the domain services to architectural solution depicted in Fig. 1) a very different approach relying on server-side and high performance big data solutions jointly with two-level workflow management systems and Science Gateways into a PaaS-based cloud infrastructure, as presented in Fig. 4.

It is worth mentioning that, with regard to the current state of the art, the architectural approach proposed by INDIGO aims at providing a set of core components still missing in the climate scientists research eco-system. It overcomes both (i) the current limitations regarding

client-side data analysis, sequential data analysis, static deployment approaches, low performance, etc. and (ii) a complete lack of workflow support, high performance and domain-oriented big data approaches/frameworks to enable large scale climate data analysis experiments.

4.4.2 Experiment design

The case study on climate model inter comparison data analysis proposed in INDIGO addresses the following data analysis classes: trend analysis, anomalies analysis, and climate change signal analysis. We started focusing our attention on the trend analysis class, as it allows validating general infrastructural aspects shared by the other two classes too.

Precipitation trend analysis has received notable attention during the past century due to its relations with global climate change stated by the scientific community. For this reason, a number of models for this atmospheric variable have been defined. To better understand model accuracy of the phenomena, firstly the results obtained by each model have to be compared against historical data to obtain possible anomalies (sub-workflows in Figure 3). Then, the anomalies have to be compared among the models (ensemble analysis) to score them and, hence, provide a final output of the experiment.

Figure 3 shows the workflow to analyze precipitation trend over a given spatial domain by comparing anomalies related to a number of models in the context of CMIP5 Federated Archive. Next this workflow is referred as the *experiment*. The experiment consists of a number of sub-workflows, which can be executed in parallel, followed by a final workflow performing an ensemble analysis. Each sub-workflow is associated with a specific climate model involved in the CMIP5 experiment. A scenario must be also defined as input.

The sub-workflow aims at performing the following tasks: (i) discovery of the two datasets (historical and future scenario data); (ii) evaluation of the precipitation trend for both the datasets separately; (iii) comparison of the trends over the considered spatial domain; and (iv) 2D map generation.

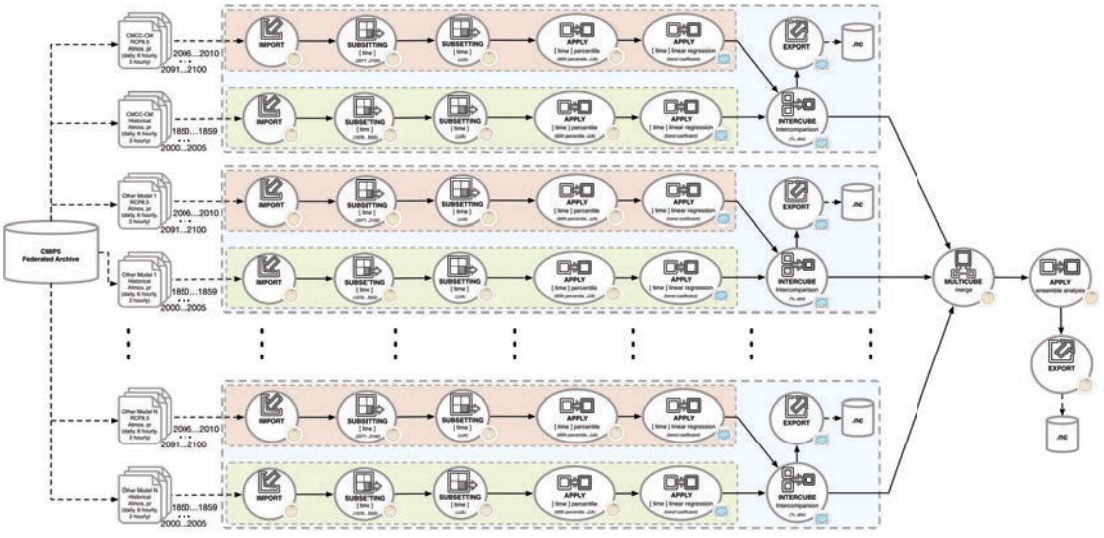

Figure 3: Design of the precipitation trend analysis experiment

The ensemble analysis, at the end of the workflow, includes the following three steps: (i)

data gathering; (ii) data re-gridding; and (iii) ensemble analysis.

For page limit issues we do not delve into the detail of the subworklow at the task-level, providing in this work only a general overview about the main parts of the experiment.

4.4.3 Running a climate data analysis experiment in INDIGO

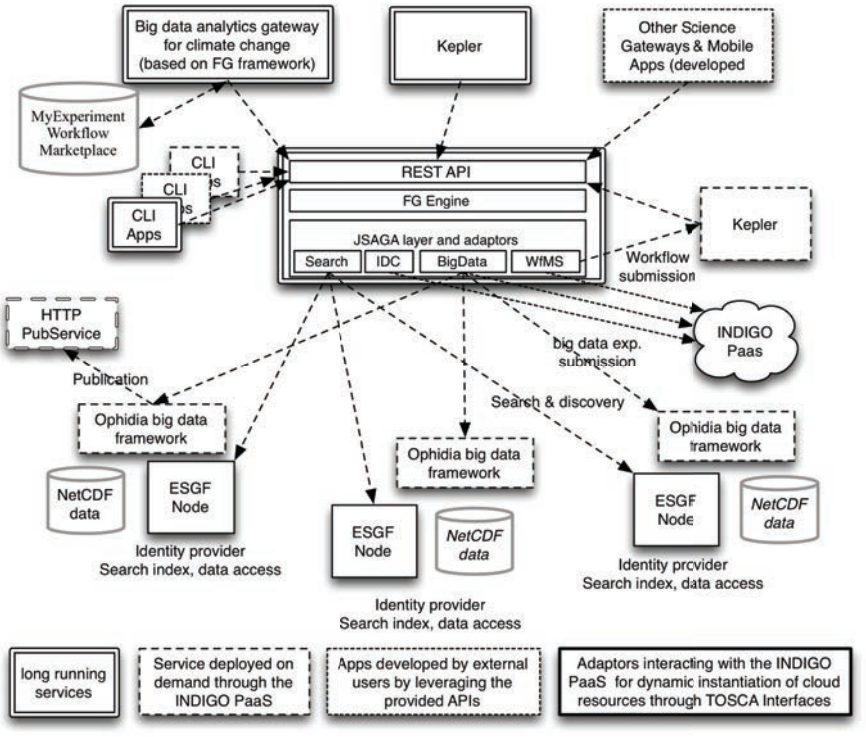

Figure 4: Mapping Climate Change use case to INDIGO-DataCloud overall architecture

During the experiment definition phase, the user will interact with the system through the portlets provided by the Data Analytics Gateway for Climate Change, choosing a specific analysis (associated to a workflow) and defining the input parameters, infrastructure/resources constraints. The user will be also able to customize predefined Kepler workflows. The workflows will be also available on a Market Place (e.g. MyExperiment) to address re-usability through a community-based approach. During the experiment run, a FutureGateway REST API invocation will be needed to submit the workflow experiment to the infrastructure. The request will be then managed by the FutureGateway Engine and dispatched to the JSAGA adaptor for workflow management. During the execution, the workflow management system instance (dynamically instantiated on the fly through the interaction with the WP5 PaaS or already statically deployed) will invoke again the FutureGateway REST API for the tasks orchestration. Specific tasks could relate to big data analytics workflows (fine grain); they will be executed through the proper JSAGA adaptor (provided by the middle tier), which will be responsible for submitting the request to the proper big data analytics engine (e.g. Ophidia) already available or dynamically instantiated on the fly in a private cloud environment through

the interaction with the INDIGO PaaS. Due to the data locality, more than one big data engine could be required during the same experiment/workflow (e.g. to run an ensemble analysis) and so a data reduction task could be also needed. To enable this scenario, a specific JSAGA adaptor for data movement will be invoked to gather on a single site the partial results obtained in parallel, on multiple sites, during the first phase of the workflow. The use case will exploit the INDIGO capabilities in terms of software framework deployed on cloud, as well as the two-level workflow strategy based on Kepler and Ophidia to run geographically distributed, scientific data analysis. More specifically:

- the general-purpose Kepler workflow management system is exploited in this use case to orchestrate multi-site tasks (level 1) related to the multi-model part of the experiment;
- the Ophidia framework is adopted at the single-site level to orchestrate the site-specific analytics workflow (level 2), related to the single-model parts of the experiment. Such workflow will run on multiple sites and will include tens of data processing, analysis, and visualization operators in Ophidia, acting at the same time as a single level-1 task in Kepler.

5 Conclusions and future work

In this paper, we have presented the ongoing work performed under the umbrella of the INDIGO-DataCloud project. The work focuses on the development of open source PaaS solution targeted at scientific communities. Results of the research will be deployable on multiple hardware, and provisioned over hybrid e-Infrastructures. We have emphasizsed part of the work related to new capabilities of applying dynamic workflow execution and support for the frameworks like Kepler or Ophidia in the field of big data analytics.

The proposed INDIGO architectural solution aims at addressing specific requirements like those of the Climate Change community by tackling the current limitations and thus enabling large scale, high performance experiments (e.g. climate data analysis). In summary, INDIGO aims at providing a core part still missing in the current scientists' research eco-system. The use case is going to be implemented on a real geographically distributed testbed involving two ESGF sites, the Euro-Mediterranean Center on Climate Change (CMCC) and the Lawrence Livermore National Laboratory (LLNL). The test case will relate to climate change datasets in NetCDF format [24], Climate and Forecast (CF) convention compliant, from the CMIP5 experiment and will be validated by a team of scientists from the two institutions. Preliminary insights about the first implementation are very promising, and mainly relate to the execution of the level-2 part of the precipitation trend analysis experiment. While the first official INDIGO release is due by July 2016, the first prototype of the climate change use case (including level-1 part) is planned to be available for testing by April 2016.

6 Acknowledgment

This work has been co-funded by the Horizon 2020 Framework Programme through the INDIGO-DataCloud Project, RIA-653549.

References

[1] CMIP5. http://cmip-pcmdi.llnl.gov/cmip5/. Accessed: February, 04. 2016.

[2] Climate Data Operators. https://code.zmaw.de/projects/cdo. Accessed: February, 04. 2016.

[3] EGI website. http://www.egi.eu/. Accessed: February, 04. 2016.

[4] EUDAT website. http://www.eudat.eu/. Accessed: February, 04. 2016.

[5] Helix-Nebula website. http://www.helix-nebula.eu/. Accessed: February, 04. 2016.

[6] INDIGO-DataCloud website. https://www.indigo-datacloud.eu/. Accessed: February, 04. 2016.

[7] The JSAGA website. http://grid.in2p3.fr/jsaga. Accessed: February, 04. 2016.

[8] The Liferay portal framework. http://www.liferay.com. Accessed: February, 04. 2016.

[9] The SAGA OGF Standard Specification. http://www.ogf.org/documents/GFD.90.pdf. Accessed: February, 04. 2016.

[10] V. et al Ardizzone. A european framework to build science gateways: Architecture and use cases. In *Proceedings of the 2011 TeraGrid Conference: Extreme Digital Discovery*, TG '11, pages 43:1–43:2, 2011.

[11] P. Baumann, A. Dehmel, P. Furtado, R. Ritsch, and N. Widmann. The multidimensional database system rasdaman. *SIGMOD Rec.*, 27(2):575–577, June 1998.

[12] Miguel Caballer, Carlos De Alfonso, Fernando Alvarruiz, and GermáN Moltó. Ec3: Elastic cloud computing cluster. *J. Comput. Syst. Sci.*, 79(8):1341–1351, December 2013.

[13] Hindman et al. Mesos: A platform for fine-grained resource sharing in the data center. In *Proceedings of the 8th USENIX*, NSDI'11, pages 295–308, 2011.

[14] Li Liu et al. A survey on workflow management and scheduling in cloud computing. In *14th IEEE/ACM, CCGrid 2014, Chicago, IL, USA, May 26-29, 2014*, pages 837–846, 2014.

[15] Luca Cinquini et al. The earth system grid federation: An open infrastructure for access to distributed geospatial data. *Future Generation Computer Systems*, 36:400 – 417, 2014.

[16] M. Płóciennik et al. Approaches to distributed execution of scientific workflows in kepler. 2013.

[17] S. Fiore et al. Ophidia: Toward big data analytics for escience. *Procedia Computer Science*, 18:2376 – 2385, 2013. 2013 International Conference on Computational Science.

[18] V. Ardizzone et al. The decide science gateway. *Journal of Grid Computing*, 10(4):689–707, 2012.

[19] S. et al Fiore. A big data analytics framework for scientific data management. In *Big Data, 2013 IEEE International Conference on*, pages 1–8, Oct 2013.

[20] Keith Jeffery Lutz Schubert. Advances in clouds. expert group report. Technical report, 2012.

[21] Members of INDIGO DataCloud collaboration. Indigo datacloud - initial requirements from research communities. Technical report, 7 2015.

[22] European Strategy Forum on Research Infrastructures. Strategy report on research infrastructures. roadmap 2010. Technical report, 2011.

[23] C. et al. Palazzo. A workflow-enabled big data analytics software stack for escience. In *HPCS, 2015 International Conference on*, pages 545–552, July 2015.

[24] R. K. Rew and G. P. Davis. 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. pages 33–40, 1990.

[25] Michael et al Stonebraker. The architecture of scidb. SSDBM'11, pages 1–16, Berlin, Heidelberg, 2011. Springer-Verlag.

[26] N. Wilkins-Diehr, D. Gannon, G. Klimeck, S. Oster, and S. Pamidighantam. Teragrid science gateways and their impact on science. *Computer*, 41(11):32–41, Nov 2008.

[27] Nancy Wilkins-Diehr. Special issue: Science gateways, common community interfaces to grid resources. *Concurrency and Computation: Practice and Experience*, 19(6):743–749, 2007.

[28] C. S. Zender. Analysis of self-describing gridded geoscience data with netcdf operators (nco), environmental modelling and software. pages 1338–1342, 2008.