Programming ecological niche modeling workflows in the Cloud

Daniele Lezzi*, Roger Rafanell†, Erik Torres‡, Renato De Giovanni§, Ignacio Blanquer† and Rosa M. Badia*, †
*Department of Computer Sciences
Barcelona Supercomputing Center, Barcelona, Spain
Email: {daniele.lezzi, roger.rafanell, rosa.m.badia}@bsc.es
†Artificial Intelligence Research Institute (IIIA), Spanish National Research Council (CSIC)
‡Instituto de Instrumentacion para Imagen Molecular (I3M), Centro mixto CSIC
§Universitat Politecnica de Valencia - CIEMAT, Valencia, Spain
Email: ertorser@upv.es, iblanque@i3m.upv.es
‡Centro de Referencia em Informaao Ambiental, Campinas, SP, Brasil
Email: renato@cria.org.br

Abstract—In the last decades biology scientists have relied on their own resources and tools to run the experiments and store the results of the analysis. However, the explosion of big data and the growing availability of computational methods find an obstacle in the lack of computational and storage resources. Cloud computing platforms are emerging as potential solution to overcome these limitations, but adaptation of the applications to enable scientific users to benefit from resources acquired on demand is a complex process requiring multidisciplinary expertise.

The EUBrazilOpenBio initiative is implementing an e-Infrastructure that provides biodiversity community with a rich set of computational and data resources exploiting existing cloud technologies from EU and Brazil. This paper presents the implementation of one of the two use cases selected, the environmental niche modeling by means of implementing such workflow through the COMPSs framework and its deployment on the EUBrazil OpenBio platform. The proposed approach has been evaluated on a Cloud testbed managed by the VENUS-C middleware.

Index Terms—Cloud; niche modeling; programming models; workflows

I. INTRODUCTION

Cloud computing has emerged as suitable model for the provisioning of resources to those scientific communities that are typically excluded by the access to supercomputing infrastructures or whose experiments require a variable demand of resources thus enabling a reduction of maintenance costs. Recently, several initiatives have proposed frameworks and services to foster the uptake of clouds for the execution of scientific applications resulting in the creation of service oriented components following different models (IaaS, PaaS, SaaS). The VENUS-C project [1] implemented a user-centric approach to the cloud, putting the requirements of end-user communities at the forefront of development, and providing scalable and interoperable cloud resources that combine both open source and commercial solutions to offer the best of both worlds.

VENUS-C provided the so called "long tail" of science with a generic cloud platform suitable for the execution of applications spanning diverse disciplines covering an existing user community of above 5,000 users.

The VENUS-C model has been adopted by the EUBrazilOpenBio initiative as one of the building blocks of the computational and data infrastructure designed to serve the needs and requirements of the biodiversity scientific community. The EUBrazilOpenBio e-Infrastructure is built by leveraging primarily on resources (textual publications and datasets, maps, taxonomies, tools, services, computing and storage capabilities) provided by Brazilian and European e-Infrastructures sustained by existing projects and initiatives. In particular, the programming models layer is an important contribution of the VENUS-C project to the scientific community. In conjunction with data access mechanisms, the programming frameworks developed in VENUS-C have proven to provide researchers with a suitable abstraction for scientific computing on top of virtualized resources. One of these tools is COMP Superscalar [2], leveraged in VENUS-C to enable the interoperable execution of the use cases on the hybrid cloud platform. The COMPSs programming framework allows the development of scientific applications and their seamless execution on a wide number of distributed infrastructures. In cloud environments, COMPSs provides scaling and elasticity features allowing to adapt the number of available resources to the actual need of the execution.

OpenModeller [3] provides a flexible cross-platform environment to perform the main tasks related with ecological niche modelling. The software includes facilities for reading species occurrence and environmental data in different formats, as well as creating, testing and projecting models into multiple scenarios. Many algorithms are provided as plugins, allowing models to be generated using different techniques. A number of interfaces are also available, including console, command-line, GUI and Web Services.

This paper describes the design and implementation of an openModeller workflow through the COMPSs framework and a performance evaluation on the EUBrazilOpenBio infrastruc-
tue.

The rest of the paper is structured as follows: section II briefly describes the EuBrazil Architecture, section III contains the description of the COMPSs framework and of the tools to enact the execution of applications, section IV illustrates the implementation of the niche modeling workflow with COMPSs, section V analyzes the performance of the ported application and section VI concludes the paper.

II. ECOLOGICAL NICHE MODELLING

Ecological Niche Modelling (ENM) is a widely used approach to predict and to understand the distribution of species. An ecological niche can be seen as the set of ecological requirements for a certain species to survive and maintain viable populations over the time [4]. In most cases, ENMs are generated by relating locations where the species is known to occur with environmental variables that may influence its distribution, and then applying an algorithm to create the model. This method is known as correlative approach [5] and the resulting model basically tries to find a representation of the environmental conditions that are suitable for the species. Such models can be projected into different geographical regions under different environmental scenarios, making it possible to predict the impact of climate changes on biodiversity, prevent the spread of invasive species, identify geographical and ecological aspects of disease transmission, help in conservation planning, guide field surveys, among many other uses [6].

Practical problems in applying ENM are associated with intensive computational requirements when models need to be generated for a large number of species using complex modelling strategies involving several algorithms and high-resolution environmental data. In the EuBrazilOpenBio project, the second use case is related with the Brazilian Virtual Herbarium (BVH) of Flora and Fungi [7], which has its own system capable of interacting with an openModeller Web Service (OMWS) instance to carry out a standard niche modelling procedure for many plant species that are native to Brazil. Species that can be modelled by the system come from the official List of Species of the Brazilian Flora [8], which currently contains 43,284 entries. The corresponding occurrence points are retrieved from speciesLink [9] - a network that integrates data from distributed biological collections, currently serving almost 4 million records considering only plant species. For species with at least 20 occurrence points available, the standard modelling procedure involves generating individual models with five different techniques: Ecological-Niche Factor Analysis [10], GARP Best Subsets [11], Mahalanobis distance [12], Maxent [13] and One-class Support Vector Machines [14]. Besides generating the models, a 10-fold cross-validation is performed to assess model quality and a final step is needed to merge the individual models into a single consensus model, which is then projected into the present environmental conditions for Brazil.

The entire process is computing-intensive, especially when dealing with multiple species. Assuming that 30% of the Angiosperms - which is just the group of flowering plants - will have enough data to generate models, an initial estimate indicates that it would require more than 10 months to process all jobs using the computational resources currently available to the BVH (a single server running with two Intel Xeon Six Core processors at 2.53GHz, 32GB of memory and 1.7TB of storage). Moreover, the number of species, occurrence points and modelling techniques is continuously changing, requiring new models to be generated over the time.

III. THE EUBRAZILOPENBIO PLATFORM

The main goal of the EUBrazilOpenBio Project is to aggregate disparate compute and data technologies into a coherent and integrated research environment for the biodiversity community.

Figure 1 depicts the prototype architecture for the implementation of the niche modelling scenario. This architecture has been designed to simplify the access to the computing resources available in the EUBrazilOpenBio infrastructure. The access to the computing resources is managed by the VENUS-C middleware through the Programming Model Enactment Service (PMES), able to schedule the jobs requests to virtual instances provided by VENUS-C and grid nodes provided by Condor [15]; the execution of the openModeller workflows is orchestrated by the COMPSs runtime with the aim of optimizing the use of resources, as explained in details in section IV-A. The VENUS-C PMES receives the execution requests from a bridge component, the ENM Service, that has been designed as dispatcher of user’s requests received from the Virtual Research Environment (VRE) portal. The ENM Service exposes an extended OMWS interface (OMWS+ in the picture) to support multi-staging and multi-parametric experiments through COMPSs and openModeller. These extensions are backwards compatible with the original OMWS specification, allowing legacy clients to be fully supported in the new implementation and, therefore, still able of submitting experiments to the execution resources without using the graphical user interface developed by the project. The ENM Service provides a bridge between the VRE, where the services of the infrastructure are provided, and the OMWS-ext instances, which can be outside of the VRE domain (i.e. in a different network or a service that is not part of the VRE). Neither OMWS nor OMWS-ext provides user support. For this reason, the ENM Service provides additional operations for accounting the experiments submitted by users, and also copies the results and the logs of the executions to the VRE, in those cases that the execution resources are outside the VRE domain. The VRE is implemented with gCube [16]. gCube, a software framework designed to abstract over a variety of technologies belonging data, process and resource management on top of Grid/Cloud enabled middleware. Through gCube VREs, groups of users have controlled access to distributed data, services, storage, and computational resources integrated under a personalised interface.
A. Extension of the OMWS interface

The openModeller XML Scheme defines the operations, input parameters and output types of openModeller. Each operation defined with this scheme supports the execution of one simple action with openModeller, for example, create, test or project a model. When a user wants to perform several actions with the same dataset, she has to submit each operation to openModeller separately. For example, to create five models with the same species occurrence dataset using five different modelling algorithms, five different requests are needed (one per algorithm). The same occurs for experiments that create models for different species occurrences using the same modelling algorithm. In the case that the operations have dependences on one another, for example, creating a model and then use it to project a dataset of occurrences points, the user is responsible for monitoring and retrieving the results of the operation that creates the model and also for including the serialized model as input parameter in the projection operation.

Another characteristic of the openModeller XML Scheme is that it doesn’t provide support for user sessions. Instead, every operation returns a ticket that the user has to store and to use with any other subsequent operation that depends on the results of the previous operation, such as monitoring and retrieving the results of that operation. Although biodiversity scientists often use openModeller to work with only one single model, EUBrazilOpenBio is targeting the massive creation of models. One of the representative use cases is to create models for an entire database of species occurrences with more than 10000 entries, such as The Virtual Herbarium of Plants and Fungi of Brazil. Additionally, these models should be automatically tested in order to ensure a minimum level of model quality. To this end, this section presents an extension of the openModeller XML Scheme to deal with the execution of multi-staging and multi-parametric experiments through COMPSs.

The proposed extended openModeller format (OM-ext) provides four additional types to the basic types1. ExtendedExperimentRequest, ExtendedExperimentStatus, ExtendedExperimentResults and ExtendedExperimentLogs. The objective of these new types is to support the following experiment pipelines, in addition to the legacy unique-stage experiments:

- Create, test and project a model
- Create and test a model
- Create and project a model
- Test and project a model

Any of these experiments will produce only one request and job id in the system, reducing the information redundancy. The second advantage over traditional OM is that these types support multi-parametric requests per experiment containing combinations of multiple species occurrences datasets and multiple algorithms definitions.

Along with the data types, also an extension to the openModeller Web Service interface has been implemented to provide the new operations that support multi-staging, multi-parametric experiments through COMPSs. The following methods are provided as extensions: submitExtendedExperiment, getExtendedExperimentStatus, extendedExperimentResults, getExtendedExperimentLogs and cancelExtendedExperiment. OMWS-ext is backwards compatible with the original OMWS interface allowing the use of legacy methods, such as createModel, testModel or projectModel, through legacy clients.

IV. COMPOSING A ENM WORKFLOW WITH COMPSs

COMPSs is a programming framework that aims to ease the development of applications in distributed environments. The unawareness of the execution environment is not the only interesting feature of COMPSs: additionally, the framework

1The openModeller XML Scheme: http://openmodeller.cria.org.br/xml/1.0/openModeller.xsd
implements a task-based programming model, and while applications are written following the sequential paradigm, the runtime is able to detect data dependencies between tasks and exploit the inherent parallelism at task level. The COMPSs framework has been recently extended in order to support the access to Cloud providers in order to provide scaling and elasticity features allowing to adapt the number of available resources to the actual need of the execution. This is achieved through the use of connectors for IaaS offerings, namely for Amazon EC2 [17], OpenNebula [18] and for Open Cloud Computing Interface (OCCI) [19] compliant middlewares (like OpenNebula and OpenStack [20]). A specific adaptor [21] allows COMPSs to benefit from the Microsoft Azure Platform-as-a-Service (PaaS).

Interoperability is key in the development of the COMPSs framework. In the context of the VENUS-C project COMPSs has been indeed enriched with the PMES service that eases the porting and execution of the applications on hybrid cloud infrastructures. The PMES for COMPSs exposes a standard OGSA-BES [22] compliant web service interface that provides interoperability with other execution services for Clusters and Grids [23]. In this way e-Science applications can be seamlessly executed on heterogeneous infrastructures without the need of developing specific adaptors. The researcher uses a client to contact the PMES in order to submit the execution of a COMPSs application. This request is expressed through a Job Submission Description Language (OGF JSDL) [24] document containing the application name, the input parameters and data references, following the HPC-BP [25] specification. The same client allows the user to interact with the cloud storage to upload a packaged version of his code; such package contains the COMPSs binaries and configuration files needed by the application.

In EUBrazilOpenBio this features enable the execution of the use cases integrating existing Brazilian grid (compute and storage) resources based on gLite, Globus or Condor and European commercial and open source Cloud solutions provided by the VENUS-C.

A. Implementation of the ENM workflow

As is described in section III-A, the proposed extension of openModeller provides a way to automatically convert multi-stage & multi-parameter experiments into a set of single legacy operations supported by openModeller suite. This is achieved by COMPSs which orchestrates the execution generating automatically the pipeline.

The ENM workflow is composed of the following operations:

- **Convert**: converts a multi-stage and multi-parameter request into a set of single operation ones.
- **Model**: models the specie distribution from a chosen algorithm and a set of occurrence points (coordinates).
- **Test**: tests and validates the model against a set of reference occurrence points.
- **Project**: projects the specie model into an environmental layer set (layout) generating the distribution map.
- **Translate**: formats and colours the projected map into a viewable image.

When an ExtendedExperimentRequest document is received, the Convert method splits the experiment request into a set of single-operation requests, species and algorithms available in the extended document having a single request per each operation, algorithm and species.

The Model operations start computing each request thus blocking the following executions of the Test and Project operations, which depend on the Model.

The Translate operation is also enqueued and started when the projected map is available, coloring it with a provided palette and eventually converting it to a desired image format as depicted in the Fig. 2:

![ENM COMPSs workflow of 1 specie and 2 algorithms.](image)

**V. PERFORMANCE EVALUATION**

In order to evaluate the performance of the implemented application a set of experiments has been conducted on a private cloud available at BSC and managed by Emotive Cloud [26]. The cluster includes a total of 96 cores available in the following way: 4 nodes with 12 Intel Xeon X5650 Six Core at 2.6GHz processors, 24GB of memory and 2TB of storage each, and 3 nodes with 16 AMD Opteron 6140 Eight Core at 2.6GHz processors, 32GB of memory and 2TB of storage each. The nodes are interconnected by a Gigabit Ethernet network and the storage is offered through a GlusterFS distributed file system running in a replica configuration mode providing a total of 8TB of usable space. On this testbed a total of 10 quad-core virtual instances with 2GB of memory and 1GB of disk space have been created running a Debian Squeeze Linux distribution.

Eight species of the genus *Passiflora*, each one with more than 20 occurrence points, were used to test the new architecture. Models were generated using the following high resolution environmental layers from WorldClim [27]: mean diurnal temperature range, maximum temperature of warmest month, minimum temperature of coldest month, precipitation of wettest quarter, precipitation of driest quarter, precipitation of warmest quarter, precipitation of coldest quarter...
and altitude. A simplified standard procedure consisting of model creation followed by an internal model test (confusion matrix and ROC curve calculation with the same input points) and a native model projection (with the same environmental layers) followed by a final image transformation was used for each species with a set of three algorithms used by BVH (SVM, ENVDIST and ENFA) executed with a different set of parameters. The Brazilian territory served as a mask in all operations. This scenario implied a total of 46 simultaneous single-operation requests.

The aim of these tests was to validate the ENM workflow COMPSs implementation evaluating the advantage of the elasticity features of the COMPSs runtime, comparing the execution on a dynamically provided pool of virtual resources with a run on a pre-deployed and static virtual environment (Grid like scenario).

Figure 3 depicts the evolution on number of virtual machines used along the execution highlighting how the runtime of COMPSs is sensible to the load produced by the tasks adapting the number of current resources to the tasks load.

After the initial scale-up phase the load remains constant until the end of the process is reached, thus starting to free resources progressively. This is not the case of the static execution scenario (overlapped on the figure) which despite being 18.4% faster than the dynamic approach, is much more expensive from a cost point of view because of the continuous usage of idle resources.

Table V compiles the execution time and speedup of running the presented experiment on both configurations limiting the maximum number of virtual machines that the system could use. As could be observed, the speedup is moderate because the application does not offer a high degree of parallelism (Figure 2). Despite this, COMPSs reaches a good performance running on a on-demand provided environment (with an average performance loss around the 9.6%), with a mean serving time of the Cloud middleware of about 120 seconds per VM. However, the speedup is not dramatically penalized, and the resources management is generally improved reducing the overall execution costs.

<table>
<thead>
<tr>
<th>#VMs</th>
<th>#Cores</th>
<th>Cloud Time</th>
<th>Speedup</th>
<th>Grid Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>00:20:21</td>
<td>1.00</td>
<td>01:46:9</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>01:00:47</td>
<td>1.98</td>
<td>00:53:23</td>
<td>1.97</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>00:33:52</td>
<td>3.55</td>
<td>00:31:06</td>
<td>3.38</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
<td>00:25:16</td>
<td>4.76</td>
<td>00:18:03</td>
<td>5.82</td>
</tr>
<tr>
<td>10</td>
<td>40</td>
<td>00:23:57</td>
<td>5.02</td>
<td>00:19:32</td>
<td>5.38</td>
</tr>
</tbody>
</table>

Table I: Execution times of static and dynamic approach

VI. RELATED WORK

The niche modeling approach used here, among other tools for biodiversity analysis, belong to the wider field of biodiversity informatics. This field is characterized by a big number of developments in several research projects, including frameworks for the composition of workflows. The Biodiversity World project [28] developed a Web Services based Grid environment whose workflow capabilities are based on Triana [29] that provides a graphical interface to assemble tools, resources and data. Triana, similarly to COMPSs, is based on GAT to execute workflows on a variety of infrastructures. An implementation of an openModeller workflow has been provided by the project. The Kepler Workflow System [30] is another framework supporting multiple models of computation suited to distinct types of analysis (processing sensor data, integrating differential equations, etc.) and provides a graphical user interface and a runtime engine that can execute workflows either from within the graphical interface or from a command line. Kepler workflows can leverage the computational power of grid technologies as well as take advantage of Kepler’s native support for parallel processing. The BioVel project [31], similarly to EuBrazilOpenBio is supporting research on biodiversity issues providing workflows services through the Taverna workflow management system [32], a suite of tools used to design and execute scientific workflows and to aid in silico experimentation. Taverna itself does not provide support for cloud execution of the workflows but it has been used as a service deployed on a cloud by several projects.

VII. CONCLUSIONS AND FUTURE WORK

This paper analyzed the first achievements of the EU-BrazilOpenbio initiative on the support to the biodiversity community in order to provide the scientists with a computational and data infrastructure based on cloud technologies. The project adopts different technologies developed in previous projects as the VENUS-C platform for the development and porting of applications on the cloud and the D4Science [33] tools for the data infrastructure. In the course of the project, the EUBrazilOpenBio e-Infrastructure is being validated to serve relevant biodiversity scientific use cases coming by strategic Brazilian and European e-science projects and initiatives. In this work the porting of one of the use case has been presented, analyzing the implementation of a niche modeling workflow by means of the COMPSs programming framework using
openModeller as modeling tool. This implementation enables the automation of several operations used for producing, testing and projecting the models. Such composite applications are offered as a service through the availability of specific methods in an extended version of the OpenModeller Web Service. Users are able, on one hand, to enhance the offered functionalities for model generation and, on the other hand, to outsource their execution to the VENUS-C Platform and its optimization thanks to the COMPSs runtime. The COMPSs Programming Model Enactment Service acts as a bridge to different infrastructures like grids and clouds. These advanced features are made available to the users through the deployment of a graphical interface in a customized Virtual Research Environment using the gCube technology.

Beside the provision of a resilient and adaptable environment for the ecological niche modeling community, the evaluation of the workflow on a cloud testbed demonstrated that the proposed solution achieves good performance providing the typical benefits of a cloud environment as elasticity and on demand provisioning of the resources.

Future work includes the development of a connector for Condor in order to extend the execution of the use cases to the brazilian grid and the use of services provided by the gCube framework in order to dynamically retrieve the locations of the layers querying the information service.

ACKNOWLEDGMENT

This work has been supported by the Spanish Ministry of Science and Innovation (contract no. TIN2007-60625 and CSD2007-00050), and by the European Commission (grant agreement no. 288754, EUBrazilOpenbio project).

REFERENCES

[27] WorldClim - Global Climate Data http://www.worldclim.org
[29] Triana http://www.trianacode.org
[31] Biodiversity Virtual e-Laboratory http://www.biovel.eu
[33] Data Infrastructure Ecosystem for Science http://www.d4science.eu

1228