A Platform to Deploy Customized Scientific Virtual Infrastructures on the Cloud

Miguel Caballer, Damián Segrelles, Germán Moltó, Ignacio Blanquer
Instituto de Instrumentación para Imagen Molecular (I3M).
Centro mixto CSIC – Universitat Politècnica de València – CIEMAT
Camino de Vera s/n, 46022 Valencia, Spain.

Abstract—**This paper presents a software platform to dynamically deploy complex scientific virtual computing infrastructures, on top of Infrastructure as a Service (IaaS) Clouds. The platform orchestrates different services to provision the virtual computing resources. It dynamically installs the appropriate software to satisfy the requirements of a researcher, both on public and on-premise Clouds. The paper also describes two case studies to deploy complex infrastructures such as a Hadoop cluster on which to execute the BLAST biomedical application or single-node infrastructures with a very specific installation and configuration process to perform NGS sequencing. This platform promotes a better use of on-premise hardware resources of a research center by allocating the computing resources just-in-time to the specific life time of the virtual infrastructures.**

I. Introduction

The diversity computing requirements for scientific applications require complex hardware infrastructure configurations and potentially incompatible specific software requirements. In a multi-disciplinary environment different researchers typically share the same hardware. Therefore, the adequate provision and configuration of computing resources could be unaffordable or impractical due to the technical overhead of switching among configurations and the effort on the configuration and customization of the infrastructures. Under these circumstances, the use of frameworks to automate the deployment and configuration of virtual computing infrastructures is necessary. Furthermore, in the context of an economic crisis, and to reduce the carbon footprint, it is specially important to properly and efficiently manage the computing resources of a research center so that the level of service and versatility is maintained without requiring additional investments in hardware.

For that purpose, Cloud computing [1], [2] is a paradigm to rapidly provision ICT resources, mainly computing and storage, that can be customized and configured to fit a particular research activity. This enables to dynamically deploy virtual infrastructures on top of a fleet of virtual machines running on a physical hardware, when using the Infrastructure as a Service (IaaS) Cloud service model. The usage of virtualization [3] enables to increase the usage of hardware and thus reducing the investments on additional hardware. In the case of a public Cloud provider, such as Amazon Web Services (AWS)[1], Windows Azure[2] or Google Cloud[3], a pay-per-use model is employed so that users are charged for the resources consumed in terms of hours of computing, network traffic, etc. In the case of private or on-premise Clouds, tools such as OpenStack [4], OpenNebula [5] or Eucalyptus [6] enable to deploy Cloud infrastructures on top of the organization's computing hardware.

In this paper, we propose a general platform to deploy on demand customizable virtual computing infrastructures, with the precise software configuration required to perform specific research activities. This platform is able to integrate computing resources both from public and on-premise Clouds. This introduces an unprecedented degree of flexibility when compared to managing physical infrastructures to support the computing requirements of complex computational research activities.

The remainder of the paper is structured as follows. First, section II presents the novelty of our platform with respect to other existent tools in the area of dynamic deployment of virtual infrastructures. Next, section III describes the software architecture of the platform together with the underlying components employed. Then, section IV describes the usage of this platform to deploy virtual infrastructures to support two different use cases. Finally, section V summarises the benefits and drawbacks of the proposed platform and points to future work.

II. Related Work

There exist works in the literature and software tools that address the deployment of virtual infrastructures. In the next paragraphs different approaches are analyzed.

Several cloud providers, such as Amazon Web Services (AWS), provide tools to deploy virtual infrastructures. In particular, CloudFormation [7] provides developers and systems administrators with an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion.

The Nimbus project team group has developed a set of tools to deploy virtual infrastructures: the Context Broker [8], the Recontextualization Broker [9], [10] and cloudinit.d [11]. In particular, the last tool submits, controls and monitors Cloud applications. It automates the VM creation process, the contextualization and the coordination of service deployment. It supports multiple clouds and the synchronization of different "runlevels" to launch services in a defined order. Furthermore it provides a system to monitor the services that uses user-created scripts to ensure that they are running. This system checks for service errors, re-launching failed services or launching new VMs. However, it enables the contextualization of VMs using

[1]Amazon AWS: http://aws.amazon.com
[2]Windows Azure: http://www.microsoft.com/windowsazure
[3]Google Cloud: https://cloud.google.com

simple scripts, which are insufficient in complex scenarios with multiple VMs and different Operating Systems.

Another example is Apache Whirr [12]. It supports deploying clusters both on EC2 and Rackspace. The user specifies the number of instances needed and the roles they must provide. The Virtual Machine Images (VMIs) to be used have to be specified beforehand. It has been initially designed to launch Hadoop clusters, but it can be extended adding new Java classes to implement the installation and contextualization of the new roles defined. It does not enable elasticity management, so once the cluster is launched its size cannot be modified by the user.

Other interesting tool is Wrangler [13]. It enables the users to define their requirements using an XML file, and specify the scripts to be used to configure the VMs to adopt a specific role within the virtual infrastructure. In this case the scripts are stored in the wrangler coordinator node, so it does not need to be previously copied in the VMIs. But the VMIs indeed require a prior step to prepare them by installing the wrangler agent and configuring it to connect to the coordinator node.

Finally, SixSq SlipStream[4] provides a web portal to deploy and configure a set of VMs. The configuration of the nodes are made by means of a list of packages to install and a script file that can be executed in each VM, which can be parameterized to create more functional and customized scripts. It can access a large list of Cloud platforms, both public and on-premise: EC2, Azure, OpenNebula, OpenStack, OCCI, etc. The main limitation is the static selection of VMIs. In addition, it does not enable elasticity management and, therefore, once the infrastructure is deployed its size cannot be modified by the user.

One common limitation in all the analyzed systems is the usage of manually selected base images to launch the VMs. This is an important limitation because it implies that users must create their own images or they must previously know the details about software and configuration of the image selected. This limitation affects the reutilization of the previously created VMIs, forcing the user to waste time testing the existing images or creating new ones (as an example in Amazon EC2 there are thousands of AMIs). Another issue is that most of them need to use a VMI specifically prepared for their tools, requiring a specific software installed, a set of scripts prepared, etc. Another important limitation is the usage of simple scripts in the contextualization, instead of DevOps tools such as Puppet [14], Chef [15], or Ansible [16], which create (nearly) system-independent configurations. Only the Nimbus "Recontextualization Broker" uses Chef to perform these tasks. This problem is exacerbated in some tools where the configuration scripts must also be stored in the base image of the VM. Furthermore, most of the aforementioned tools do not provide an easy language nor an user friendly interface, hindering the usage for non advanced users.

The platform defined proposes an architecture which separates the management of VMIs, the infrastructure descriptions and their installation and deployment on a Cloud. It offers a flexibility that the previous aforementioned works do not offer, allowing to extend and improving the related works presented in the following aspects:

[4]https://slipstream.sixsq.com

- The platform does not exclusively focus on a specific type of infrastructure (PC cluster, grid testbed, parallel environment, etc), since it allows to deploy any kind of complex infrastructure.
- One common limitation of the previous works is the usage of pre-packaged VMIs. This is an important limitation because users are limited to use the pre-selected software of the VMIs or they must manually install the desired software after the virtual infrastructure has been provisioned. This platform, instead, provides the ability to automatically install at runtime software and data dependences in the virtual infrastructure, and perform complex configurations of the infrastructures. Therefore, it is easier to keep the infrastructure updated, since new version of packages can be installed in each new deployment.
- The platform is designed to enable reusing both the infrastructure descriptions, using a repository of high level infrastructure descriptions, and the VMIs, using a catalog with their specific metadata.
- The same complex infrastructure can be deployed both on-premise and in a public Cloud, thus enabling to outsource computation to the public Cloud when there is a shortage of computational resources on-premise.

III. Proposed Architecture

In the scenario proposed in this paper we assume that a research institution has a pool of computing hardware that is managed by a Cloud Management Platform (CMP), such as OpenNebula, in order to run virtual computing infrastructures on top of that hardware. This provides all the benefits of virtualization such as isolation and better hardware utilization, thus avoiding the usage of dedicated hardware. Many organizations are already using virtualization to leverage server consolidation [17] so that, for example, a single machine hosts different virtual machines (VMs) with separate roles such as mail server, application server, etc.

Alternatively, the scenario also considers a research center that outsources part of their computing to a public Cloud provider, in order to provision VMs for operational support of their staff. Ultimately, scientists could certainly bypass their research institution and deploy the required resources on a public Cloud by themselves, if no support is provided by their institution.

Under the aforementioned assumptions, The presented platform enables creating virtual computing infrastructures on top of public and on-premise Clouds in order to support research activities. Therefore, the main aim of the platform is to simplify the definition of virtual computing infrastructures and to perform their automated deployment, by dynamically provisioning and relinquishing the related resources, together with the appropriate configuration to fit the requirements of a certain scientific application.

Figure 1 summarises the main steps involved when interacting with the platform. First of all, the researcher must analyse the infrastructure requirements for a specific application, in

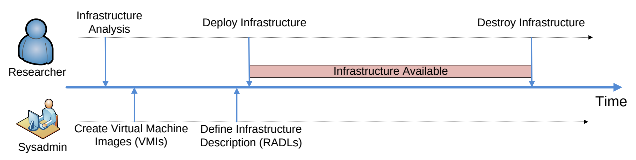


Fig. 1. Time line and main actors involved in the virtual computing infrastructures.

terms of hardware, software, and special configuration requirements. For example, a researcher could require a Hadoop cluster with N 64-bit virtual machines with Ubuntu GNU/Linux 12.10. The researcher passes the requirements to a system administrator (*sysadmin*) of the platform, which could be the researcher itself if it has the appropriate skills. The sysadmin must check if there are appropriate Virtual Machine Images (VMIs) stored in the repository and if not, he must install and configure them. Finally he must register the new VMIs in the repository and catalog service including the correct metadata (OS, applications, etc.) to enable its usage in the platform.

The sysadmin must distinguish between the software that can be dynamically deployed at runtime on the VMs and the one that should be pre-installed in the VMIs due to complex software requirements or a lengthy installation process. Then, the sysadmin describes the infrastructures using a higher level declarative language called RADL, which will be explained later on. Each description is a recipe of the hardware, software and configuration required to instantiate a given infrastructure, regardless of the underlying Cloud infrastructure to be employed. An infrastructure can be composed of one or more VMs with possibly different configurations. The usage of a high level recipe means that the same computing infrastructure can be deployed on an on-premise Cloud and a public Cloud, in case no spare hardware resources are available in the on-premise Cloud. Also, to have recipes for each infrastructure enables to automate their deployment process for different activities, such as a periodic task.

The infrastructure descriptions are made available to the researchers through a web application. Infrastructures can therefore be instantiated, which means to actually create and configure the VMs to deploy the computing infrastructure in a Cloud back-end, with a researcher's single click. This triggers all the work of the platform which is in charge of deploying the VMs out of the VMIs, by interacting with different IaaS Cloud providers, dynamically installing the specified software and providing the configuration among the VMs in order to create an available virtual computing infrastructure that satisfies the requirements specified in the high level, declarative RADL description.

Once the virtual computing infrastructure is up and running the researcher can access it via SSH or using remote desktop tools such as FreeNX, depending on the configuration specified. He will be also in charge of destroying the infrastructures in order to release the provisioned hardware resources.

Figure 2 shows the architecture of the platform. The central component is the Infrastructure Instantiator (II) that is in charge of deploying and contextualizing the VMs of the virtual infrastructures. The II uses the infrastructure descriptions, which are RADL documents stored in the Repository of

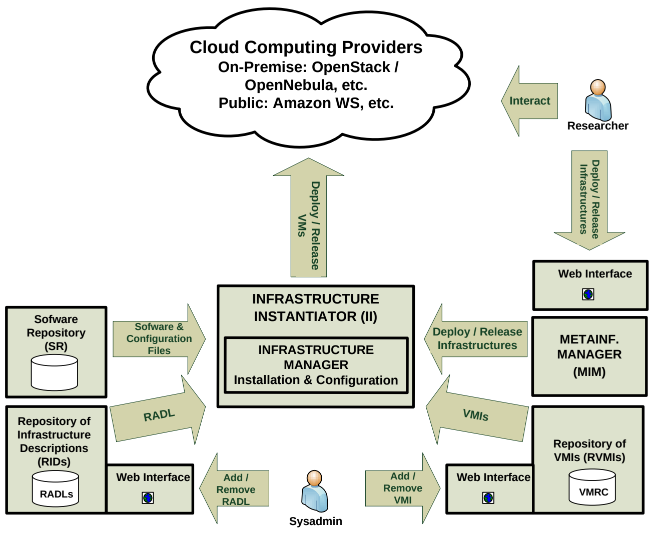

Cloud Computing Providers
On-Premise: OpenStack / OpenNebula, etc.
Public: Amazon WS, etc.
Interact
Researcher
Deploy / Release VMs
Deploy / Release Infrastructures
Web Interface
INFRASTRUCTURE INSTANTIATOR (II)
INFRASTRUCTURE MANAGER Installation & Configuration
Sofware Repository (SR)
Sofware & Configuration Files
Deploy / Release Infrastructures
METAINF. MANAGER (MIM)
RADL
VMIs
Repository of Infrastructure Descriptions (RIDs)
RADLs
Web Interface
Add / Remove RADL
Sysadmin
Add / Remove VMI
Web Interface
Repository of VMIs (RVMIs)
VMRC

Fig. 2. Main architecture of the platform.

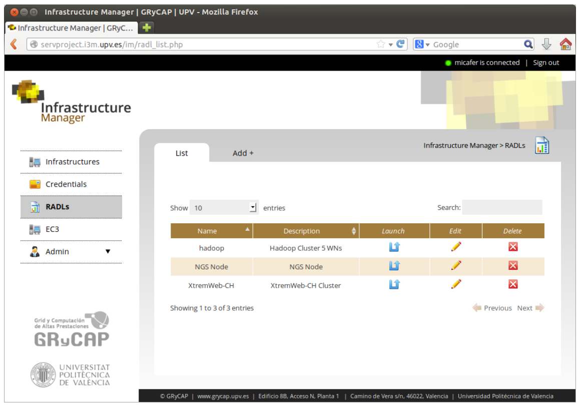
Infrastructure Descriptions (RIDs). It selects the most suitable VMIs available from the Repository of Virtual Machine Images (RVMIs), implemented by the VMRC (Virtual Machine image & Repository Catalog) [18]. Then, in the contextualization step, it uses the software and data packages from the software repositories. Finally the MetaInfrastructure Manager (MIM) controls the II by organizing and scheduling when the infrastructures will be deployed or released. The following subsections describe the main components and technologies of the platform.

A. *Infrastructure Instantiator (II)*

The Infrastructure Instantiator (II) is the component in charge of managing all the life-cycle of virtual infrastructures. It instantiates a RADL document to create the infrastructure by orchestrating all the required tasks. It can add or remove nodes dynamically to fit the size of the infrastructure to the real requirements, and apply the required configuration of the software components. It has been implemented using the Infrastructure Manager (IM) [20], [19].

The II uses the RVMIs to get the most suitable VMI considering the restrictions defined in the RADL document. Then it contacts the Cloud back-end to launch all the defined VMs and, finally, it uses Ansible to configure the whole infrastructure. The II has a modular design enabling the interoperability with the maximum number of cloud deployments. Currently it supports on-premise Cloud tools such as OpenNebula and OpenStack as well as any OCCI-compliant Cloud provider. It can also interact with Amazon Elastic Compute Cloud (EC2) to deploy VMs on that public Cloud.

In the contextualization phase (installation and configuration), all the software and data needed to fulfil the requirements of the RADL document are dynamically retrieved from the SR. The IM is implemented as a service that exports a simple interface that can be used from different client programs. Two client interfaces have been developed, a simple command line program and an API (available through XML-RPC and REST).


Infrastructure Manager | GRyCAP | UPV - Mozilla Firefox
Infrastructure Manager | GRyC...
servproject.i3m.upv.es/im/radl_list.php
Google
micafer is connected | Sign out
Infrastructure Manager
Infrastructures
Credentials
RADLs
EC3
Admin
List
Add +
Infrastructure Manager > RADLs
Show 10 entries
Search:
Previous Next
Grid y Computación de Altas Prestaciones
GRyCAP
UNIVERSITAT POLITÈCNICA DE VALÈNCIA
© GRyCAP | www.grycap.upv.es | Edificio 8B, Acceso N, Planta 1 | Camino de Vera s/n, 46022, Valencia | Universidad Politécnica de Valencia

Name	Description	Launch	Edit	Delete
hadoop	Hadoop Cluster 5 WNs	⤴	✎	✖
NGS Node	NGS Node	⤴	✎	✖
XtremWeb-CH	XtremWeb-CH Cluster	⤴	✎	✖

Showing 1 to 3 of 3 entries

Fig. 3. User Interface of the Repository of Infrastructure Descriptions

B. *Repository of Infrastructures Descriptions (RIDs)*

To describe an infrastructure, both the hardware characteristics and the software and configuration issues should be specified. All these features must be described in a formal document to enable the automation of these tasks. These documents are expressed using the Resource Application Description Language (RADL) [20], [19], which is a simple high level and declarative Domain Specific Language (DSL) to describe the hardware and software features of a virtual infrastructure (see an example in Figure 4). A RADL document specifies the requirements of each type of VM needed in the infrastructure and the number of instances of each type to be deployed on top of the IaaS Cloud.

The Repository of Infrastructures Descriptions (RIDs) stores a set of RADL documents in a web application where the user can specify a name and a description for each document. The creator of an RADL can also specify a set of permissions to access the document, enabling to share the RADLs with different groups of users with different permissions (read, write, launch). This represents a central access point to share RADL documents created by the sysadmins that can be used by the researchers using a simple web interface (as shown in Figure 3). This way, infrastructures can easily be deployed by only means of a web browser, without any special client-side configuration.

C. *MetaInfrastructure Manager (MIM)*

The MetaInfrastructure Manager (MIM) organizes and schedules when the infrastructures will be used, deploying or releasing the virtual infrastructures defined, in an automatic or manual way. This component is the bridge between the II and the researcher. The MIM interacts with the researcher through a web interface, and launches instructions to the II via its API. The web interface is also integrated with the RIDs enabling to use the same interface to access the RADL documents and launch the infrastructures (Figure 3), enabling to create a virtual infrastructure with a simple mouse click.

IV. Use Cases

This section shows the advantages of the proposed platform for two different real scientific applications with specific computing and configuration requirements. In particular, two

```
network privada
network publica (outbound = 'yes')

system front (
cpu.arch='x86_64' and cpu.count>=2 and
memory.size>=1536m and
net_interface.0.connection = 'publica' and
net_interface.1.connection = 'privada' and
disk.0.os.name='linux' and
disk.0.os.flavour='ubuntu' and
disk.0.os.version='12.04' and
disk.0.applications
    contains (name='ansible.modules.micafer/ansible-playbook-hadoop')
)

system wn (
cpu.arch='x86_64' and
memory.size>=1536m and
net_interface.0.connection='privada' and
disk.0.os.name='linux' and
disk.0.os.flavour='ubuntu' and
disk.0.os.version='12.04'
)

configure front (
@begin
---
  - tasks:
    - include: hadoop/tasks/hadoop-master.yml hadoop_master=$IM_MASTER_FQDN

@end
)

configure wn (
@begin
---
  - tasks:
    - include: hadoop/tasks/hadoop-wn.yml hadoop_master=$IM_MASTER_FQDN

@end
)

deploy front 1
deploy wn 5
```

Fig. 4. RADL document of the Hadoop Use Case.

different Cloud platforms have been used in the following tests: The first is an on-premise cloud with OpenNebula composed of three Dell blades (M600 and M610), each one with two Quad-Core processors and 16 GB of RAM. The second one uses the public Cloud infrastructure of Amazon EC2.

A. *Use Case: Hadoop cluster for Hadoop BLAST*

MapReduce [21] is a programming model widely used to process large amounts of data. MapReduce is implemented on top of Hadoop[5] which provides the implementation of a HDFS replicated and distributed file system to manage a large volume of data. Hadoop and MapReduce are widely used in Cloud computing infrastructures to tackle Big Data problems. MapReduce is appropriate for problems of large text indexing and searching, such as the genome analysis in bioinformatics. We selected the problem of alignment of sequences of nucleotides using BLAST (Basic Local Alignment Search Tool) [22], as one of the most popular tools in homology search and function analysis in biomedicine. BLAST was used in conjunction with other software developed to enable this application to run in a Hadoop cluster[6].

The IM provides a recipe to enable the configuration of a Hadoop cluster to non advanced users, who could use the RADL document shown in figure 4. This recipe deploys and configures a fully operative Hadoop cluster. The number of nodes can be easily defined by the user.

Table I shows the time spent to launch several Hadoop clusters with different configurations in both AWS EC2 and

[5]http://hadoop.apache.org
[6]http://salsahpc.indiana.edu/tutorial/hadoopblast.html

TABLE I. HADDOP CLUSTER CREATION TIMES

	6 node (ONE)	11 nodes (ONE)	11 nodes (EC2)	51 nodes (EC2)
Creation	12:05	22:13	9:29	14:26
Node Addition	5:13	5:33	5:46	11:23
Node Removal	1:18	1:32	3:17	6:36

OpenNebula (ONE). The size of the clusters was 6, 11 and 51 (1 was the front-end and the rest were of WNs). In EC2, the instance type selected by the IM was "m1.small", as the only requirement specified in the RADL is to have more than 1.5 GB of RAM. For the front-end node, IM selected a "c1.medium" as it is the cheapest one with two cores and 1.5 GB of RAM.

In OpenNebula, deployment times increases with the number of VMs, since the on-premise Cloud platform suffers from a bottleneck in the network and disk access. This effect does not show up in EC2.

It can be seen that the tasks related with the installation of Ansible together with the booting and the removing of nodes takes longer in EC2, since preparing and configuring the VMs require connection to the master Ansible service located at UPV. The on-premise OpenNebula Cloud is in the same local network as the Ansible service, whereas the instances in AWS EC2 require an external connection. If necessary, these steps can be sped-up by deploying an Ansible master in another EC2 instance. On the contrary, the configuration step is faster in the public Cloud due to the features of the underlying hardware and the bottlenecks of the on-premise deployment. This behaviour is similar in all the use cases.

B. Use Case: NGS: Next Generation Sequencing

The analysis of mutations consists on identifying significant variants of an individual's genome with respect to the consensus reference genome and the existing known mutation databases. With the advent of massive sequencing (also known as Next Generation Sequencing - NGS), this process has increased in complexity and resource requirements. There are a wide range of tools used by the researchers for the preprocessing, alignment to reference, identification and variants and searching of variants in data bases which constitute well-known processing pipelines. Those tools require local copies of the reference genome and the individual's sequences.

The use case implemented integrates FastQC[7] for quality analysis, Bowtie2[8] for the alignment, Samtools[9] for sorting, data conversion and pile-uping, GATK[10] for the computation of Variant Calls and Galaxy[11] as interface.

This use case shows how to integrate a set of the necessary tools in a VM to provide a scientist with a framework for research. By properly defining the recipes the users guarantees that a specific version of the tools are used. This increases repeatability and could enable reviewers of scientific publications to deploy exactly the same environment to verify the results presented in an article. The use of the IM service is free and the deployment can be done in any public Cloud or on-premise infrastructure.

NGS analysis may have different requirements in terms of the number of cores and memory at different steps of the processing pipeline. This approach may be used to use local on-premise resources as much as possible and to scale-out to public larger resources when instances with larger needs are required. For example, 64GB instances may not be available on a local infrastructure but they may not be necessary for all the experiments, only for large genomes or special configurations.

Creation time has been of 19:55 in ONE and 9:20 in AWS EC2. The experiment used short read sequencing of the chicken and its reference genome from UCSC[12], which is a 319MB file, and focusing only on chromosomes 7, 8 and 9.

C. Analysis of Deployment

The functionality of the platform has been tested with two representative use cases, a Hadoop cluster for Hadoop BLAST and Next Generation Sequencing. In each use case a specific infrastructure with a complex configuration has been deployed. The platform have been able to deploy the very same infrastructure in different Clouds (private and public) and scale the infrastructure in a ease way to the final user, demonstrating one of the main features of the platform, the ability to deploy infrastructures on different Clouds with the same RADL definition.

The results show that in the order of minutes (around 15) a Hadoop cluster of 50 nodes can be deployed on a Cloud infrastructure. Also, the addition of new nodes is relatively fast operation with about 11 minutes. Finally the removal operation is the quickest operation using about 3 minutes. Also, the results show that almost 20 minutes are required to deploy and configure the whole infrastructure needed to perform the second use case.

In both cases, the platform enables a quick deployment of the infrastructure in case of needing to deploy them to support scientific experiments. The times includes all the steps required to deploy the VMs, install Ansible in one node (designated as master node), then use it to contextualize all the deployed VMs with the user requirements expressed in the RADL document.

V. CONCLUSION AND FUTURE WORK

This paper has presented a software platform to easily create scientific on demand virtual computing infrastructures. Firstly, it should be pointed out that creating a high level description or recipe (using RADL) of the desired infrastructure is a much more efficient way to provision infrastructures when compared to the traditional way (provision the computational resources, manual installation of software and configuration, etc.). Secondly, this enables repetition and automation. The same infrastructure can be deployed as many times as required. Even more, the RADL description can be reused and shared so that the definition of other infrastructures can build up on the recipes of previously defined infrastructures.

[7]http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
[8]http://bowtie-bio.sourceforge.net/bowtie2/
[9]http://samtools.sourceforge.net
[10]http://www.broadinstitute.org/gatk/
[11]http://galaxyproject.org

[12]http://hgdownload.soe.ucsc.edu/goldenPath/galGal4/bigZips/

Thirdly, the ability of the platform to interact with different IaaS Cloud backends is of special importance. Notice that the same RADL definition serves to create the infrastructure on a private Cloud (such as an OpenNebula-based Cloud deployment) or in a public Cloud such as Amazon EC2. This makes it very easy to perform Cloud bursting, where a Cloud provider is employed in case no spare hardware resources are available in the on-premise Cloud of the research institution. Therefore, there is no need for an institution to over-provision hardware resources to cope with sudden workload peaks, where many researchers are simultaneously requesting computing resources to deploy their virtual infrastructures, since it can outsource the deployment of some virtual infrastructures to a third-party Cloud provider.

Notice that the usage of dynamically deployed infrastructures enables always to start off with a fresh installation. This means that all the users' data and modifications are lost when the infrastructure is torn down. Any unauthorized modification of the infrastructure (even malware, rootkits, etc.) will be vanished when the infrastructure's resources are relinquished. This provides a safe, pristine environment every time an infrastructure is deployed.

The usage of such a platform enables research centers to better take advantage of existing hardware resources. By leveraging the elasticity of the Cloud platform, in terms of rapidly provisioning resources, and the advantages of virtualization, such as isolation and multi-tenancy, the proposed platform paves the way for a versatile creation of scientific infrastructures.

Since the platform API and tools can be considered stable, future works involve improving the web applications of the platform in order to enhance the user interaction for the less experienced users. This opens avenues to democratize the access to computing resources on demand to support research activities, even for scientists that belong to areas different from computer science.

References

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing. NIST Special Publication 800-145 (Final)," Tech. Rep., 2011. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[2] R. Buyya, J. Broberg, and A. M. Goscinski, *Cloud computing: Principles and Paradigms*. Wiley, 2011. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/9780470940105.fmatter/pdf

[3] J. Sahoo, S. Mohapatra, and R. Lath, "Virtualization: A Survey on Concepts, Taxonomy and Associated Security Issues," in *2010 Second International Conference on Computer and Network Technology*. IEEE, 2010, pp. 222–226. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5474503

[4] OpenStack, "OpenStack," 2013. [Online]. Available: http://openstack.org

[5] B. Sotomayor, R. Montero, I. Llorente, and I. Foster, "Capacity leasing in cloud systems using the opennebula engine," *Cloud Computing and Applications*, vol. 2008, pp. 1–5, 2008.

[6] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The Eucalyptus Open-source Cloud-computing System," in *Proceedings of 9th IEEE International Symposium on Cluster Computing and the Grid*, 2009.

[7] Amazon Web Services, "AWS CloudFormation," 2013. [Online]. Available: http://aws.amazon.com/es/cloudformation/

[8] K. Keahey and T. Freeman, "Contextualization: Providing One-Click Virtual Clusters," in *Fourth IEEE International Conference on eScience*, 2008, pp. 301–308.

[9] ——, "Architecting a Large-Scale Elastic Environment: Recontextualization and Adaptive Cloud Services for Scientific Computing," pp. 409–418, 2012.

[10] P. Marshall, K. Keahey, and T. Freeman, "Elastic Site: Using Clouds to Elastically Extend Site Resources," in *Proceedings of the 2010 IEEE/ACM 10th International Conference on Cluster, Cloud and Grid Computing*, ser. CCGRID '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 43–52.

[11] J. Bresnahan, T. Freeman, D. LaBissoniere, and K. Keahey, "Managing appliance launches in infrastructure clouds," in *Proceedings of the 2011 TeraGrid Conference: Extreme Digital Discovery*, ser. TG '11. New York, NY, USA: ACM, 2011, pp. 12:1—-12:7.

[12] Apache, "Whirr," 2013. [Online]. Available: http://whirr.apache.org/

[13] G. Juve and E. Deelman, "Automating Application Deployment in Infrastructure Clouds," in *Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science*, ser. CLOUDCOM '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 658–665.

[14] Puppet, "Puppet Labs: IT Automation Software for System Administrators." [Online]. Available: http://www.puppetlabs.com

[15] Opscode, "Chef," 2013. [Online]. Available: http://www.opscode.com/chef/

[16] M. DeHaan, "Ansible," 2013. [Online]. Available: http://ansible.cc/

[17] W. Vogels, "Beyond server consolidation," *Queue*, vol. 6, no. 1, p. 20, Jan. 2008. [Online]. Available: http://dl.acm.org/ft_gateway.cfm?id=1348590&type=html

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

[19] M. Caballer, I. Blanquer, G. Molto, and C. de Alfonso, "Dynamic management of virtual infrastructures," *Journal of Grid Computing*, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10723-014-9296-5

[20] C. de Alfonso, M. Caballer, F. Alvarruiz, G. Moltó, and V. Hernández, "Infrastructure Deployment Over the Cloud," in *2011 IEEE Third International Conference on Cloud Computing Technology and Science*. IEEE, Nov. 2011, pp. 517–521. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6133186

[21] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008. [Online]. Available: http://doi.acm.org/10.1145/1327452.1327492

[22] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool." *Journal of molecular biology*, vol. 215, no. 3, pp. 403–410, Oct. 1990. [Online]. Available: http://dx.doi.org/10.1006/jmbi.1990.9999