A Cloud framework for Problem-Based Learning on Grid Computing

J. Damian Segrelles Quilis*, German Moltó, Ignacio Blanquer

Instituto de Instrumentación para Imagen Molecular (I3M), Universitat Politècnica de València (UPV), Camino de Vera S/n, 46022 Valencia, Spain

Abstract

Training on Grid technologies have traditionally used existing Grid infrastructures to implement the hands-on education activities. However, these infrastructures are insufficient to develop all training skills as they can only be employed for the development of Grid applications, and they are limited for learning the management and configuration of Grid resources. The paper presents a set of educational activities grouped on a Project Based Learning (PBL) framework for training on Grid technologies. A Cloud-based tool has been implemented to provide Grid infrastructures as a Service on the cloud, with enhanced scalability and administration capabilities. The PBL has achieved a high impact in the teaching-learning process, addressing the training in all the necessary skills and efficiently providing Grid infrastructure resources on public clouds at a moderate cost. Finally, we evaluated the students' opinion on the activities achieving a very satisfactory result and a reasonable balance on the complexity of the PBL stages.

Keywords: Grid computing, Cloud computing, Project Based Learning

1. Introduction

Grid computing [1] is a large-scale geographically distributed hardware and software infra-structure composed of heterogeneous networked resources owned

*Corresponding author
Email address: dquilis@dsic.upv.es (J. Damian Segrelles Quilis)

and shared by multiple administrative organizations which are coordinated to provide transparent, dependable, pervasive and consistent computing support to a wide range of applications. These applications can perform either distributed computing, high throughput computing, on-demand computing, data-intensive computing, collaborative computing or multimedia computing. During the last two decades, Grid Computing teaching [2] has been fundamental in many areas of knowledge, especially in High Energy Physics [3], Biomedicine [4], Astrophysics [5] and different engineering areas [6, 7], fulfilling the high computational and storage demand of resources from such challenges. One key aspect in teaching the basis of Grid Computing is the concept of *Virtual Organisation* (VO), [8] which is defined as a dynamic set of end-users (*Grid Users*) and/or institutions, sharing a set of computational or storage resources (*Grid Infrastructure*) and defining resource-sharing rules. The VOs share common targets and requirements, but may vary in size (users and resources), scope, duration and structure. Teaching Grid Computing is not an easy process, as most of the *Grid Users* (e.g. physicists, astrophysics, engineers) lack skills related to distributed computing technologies. For that, the *Grid Middlewares* (e.g. Globus Toolkit [9] or UMD [10, 11]) provide *Grid Users* with access to the resources and services (*Grid Infrastructure*) that support the VOs to which they belong. Grid Middlewares hide the complexity of managing distributed and heterogeneous resources through standard interfaces, thus easing the development and execution of applications that tackle scientific or engineering challenges requiring a large capacity computing [12]. However, *Grid Middlewares* are difficult to use and managing *Grid Infrastructures* still requires learning different roles:

- *Grid Users*, who develop advanced Grid applications and perform scalability studies.
- Administrators of a VO (*VO Admin*), who manage the membership (authentication and authorisation) of *Grid Users* in a VO.
- Resource administrators of an administrative domain (*Domain Admin*),

who manage the certification of resources and users of an institution supporting a VO.

- Resource Managers (*Resource Admin*), responsible for configuring the Grid services.

The Universitat Politècnica de València (UPV) offers the Master's Degree in Cloud Computing and High Performance Computing (MUCNAP)[1], where the subject *Grid and Cloud Computing Concepts* (GCCC) instructs students in the roles of *Grid Users*, *VO Admin*, *Domain Admin* and *Resource Admin roles.* This work presents the design of a set of Educational Activities (EAs) in the framework of a Project Based Learning (PBL) [13] methodology that has been implemented in GCCC the last 3 years. Also, a novel cloud-based tool named *Grid as a Service* (GaaS) is presented. This tool is an educational resource that enables the design of these new EAs, providing a more complete teaching experience for training on the aforementioned roles.

After this introduction, the rest of the paper is structured as follows. First, section 2 (background) justifies the need to design new EAs in the aforementioned master course to deal with all the learning objectives aimed at the Grid computing course [14]. This section also describes the need of creating a new cloud-based tool as a learning resource for enabling these EAs, substituting other learning resources currently used. Next, section 3 (related work) describes existing learning resources for training on Grid Computing, highlighting their advantages and how the cloud-based tool developed in this work provides better performance, in terms of learning objectives accomplished, to train Grid computing. The fourth section outlines the PBL methodology designed for the GCCC course, describing the learning objectives, teaching methods and the required resources for performing each EA. Next, section 5 introduces the GaaS tool and describes how it fits in the PBL methodology. Section 6 first presents

[1]Master's Degree in Cloud Computing and High Performance Computing: `https://www.upv.es/titulaciones/MUCPD`

the student's opinion of the PBL experience and a statistical analysis about the complexity of the PBL stages and activities designed. Also this section evaluates the Cloud-based tools in terms of deployment times and economical costs. Finally, the last section presents the conclusions and further works.

2. Background

During the first years of GCCC, the EAs implemented were exclusively focused on how to use the Grid infrastructure from the point of view of the *Grid User* role, training students to: a) Create, execute and monitor basic Grid applications during their whole lifecycle, from the creation and submission of Grid jobs until the gathering of the final results; b) the data transfer and storage management in Grid applications; and c) the discovery and monitoring of resources provided by a Grid infrastructure. For this purpose, we used production Grid platforms and VOs, such as the *tut.vo.ibergrid.eu* VO and the Grid INFN Virtual Laboratory for Dissemination Activities (GILDA) [15]. These EAs were individual tasks where each student performed activities to achieve specific learning objectives related only to the *Grid User* Role. These learning resources were supported on production large-scale Grid Infrastructures, such as the Spanish National Grid Initiative (Es-NGI)[2] and the European Grid Infrastructure (EGI)[3].

Despite the benefits of using production infrastructures, they were insufficient to train students on all the skills of Grid technologies, since production infrastructures can only be used to partially accomplish the learning objectives related to the basic development of Grid applications. No learning objective related to the configuration and management of Grid resources and services and VO administration could be trained. Also, despite the fact that production Grid infrastructures gather a large amount of resources, they have limited computing and storage resources dedicated to training VOs, which prevents the study of

[2]Spanish national GRID initiative: `http://www.es-ngi.es`

[3]European Grid Infrastructure: `https://www.egi.eu/`

the scalability of advanced Grid applications by *Grid Users*. Moreover, those infrastructures are pre-defined and managed by external institutions, and thus, it is not possible to train the *VO Admins* roles or the security issues regarding the authentication/authorisation. Finally, the resources and services offered were also managed by external institutions, impeding training the *Resource Admins* or *Domain Admins* roles.

To cover the mentioned training deficiencies, a set of new EAs were designed and included in a PBL framework. The new EAs include the advanced development of Grid applications and the management and configuration of Grid resources and services, as learning objectives of the course. To implement these new EAs, an innovative cloud-based tool named Grid as a Service (GaaS) was built, replacing the Grid Infrastructures as learning resources. GaaS is a cloud-based tool that allows to build virtualized *Grid Infrastructures* on-demand where all the resources and services of a VO are under the control of the students and the teacher, to implement the activities focused to train the roles of *VO Admins*, *Domain Admins*, *Resource Admins* or even scalability experiments on the *Grid infrastructure* to test advanced Grid applications.

3. Related Works

As indicated above, there have been initiatives at European level to build Grid platforms [15, 16] and VOs [17, 18] for training Grid Computing, even beyond the European level, in collaborations between Europe and Latin America [19]. However, the *Grid Infrastructure* for these VOs and platforms are restricted in terms of access, scalability and administration, and thus it is not possible to perform activities for *Admins roles* or advanced *Grid Users*.

Other activities have built specific tools aimed at virtualized Gid infrastructures such as GRIDSEED [20] or GridBuilder [21]. GRIDSEED is based on virtual machines and can be used to easily deploy a training Grid infrastructure. The problem of this tool is that it requires manual intervention to scale up and down the Grid Infrastructure and also requires on-premises hardware to deploy the virtual machines. GridBuilder is a web-based virtual machine manager that supports the rapid creation and customisation of virtualised Grid nodes based on standard configurations allowing to scale up the *Grid Infrastructure*. Both GRIDSEED and GridBuilder demonstrated that virtualisation is a good solution to create on-demand *Grid Infrastructures*. However, they were based in VMs which must be installed on premises and thus they have scalability limitations. In this work, we evolve this idea by moving the virtualised infrastructures to cloud providers. Cloud technologies are appropriate to create *Grid Infrastructures as a Service* to carry out EAs to train Grid computing concepts since they allow to dynamically and elastically provisioning virtualised infrastructures to create VOs and scale up and down the resources on-demand either using public cloud providers (such as Google cloud, Microsoft Azure or Amazon Web Services - AWS) and on-premises Cloud Management Platforms (such as OpenNebula or OpenStack).

In this work, we present the design of a PBL framework where a group of students work together to create a VO and the supporting services and infrastructure. To design the EAs of the PBL we have considered the best practices for Grid computing teaching in [19] and a virtualised Grid infrastructure deployed on a cloud provider managed by the Grid as a Service (GaaS) cloud-based tool.

4. Project based Learning Framework

This section describes how the Project Based Learning has been designed. As a first step, the problem to solve in the PBL has been defined and the Learning Objectives (LO) to address have been listed. Then, the EAs to carry out and the timing dependencies between them have been designed. Finally, each EA has been described: a) what a student learns; b) portfolio evidence to be assessed and required resources.

As stated in the introduction, Grid computing [1] is the use of widely distributed computer resources to reach a common goal. It is based on the concept of *Virtual Organisation* (VO). A VO is a dynamic set of end users (*Grid User* role) who have common access permissions and institutions (*Administrative Domain* role and *Resource Admin* role) which share resources. *Grid Users* develop and run Grid applications to tackle high capacity problems that require high computational and storage requirements gathered by differents *Administrative Domains* provided by research centers and universities (*Resource Admin* role). Thus, the learning framework of GCCC has been designed around the VO concept, implementing a Project Based Learning (PBL) [22] framework. PBL is a student-centered approach in which students learn about a subject by working in groups to solve a problem. In particular, in this work all students form a single working group to learn the basis of Grid Computing during the course and collaborate together to build a textit Target VO as the final problem. The design of the PBL has taken into account:

- The PBL framework has to be organised in phases (learning modules) that develop the fundamental aspects of the best practices for teaching Grid Computing [14], which are: a) Grid Security; b) Grid Job Management; c) Grid Data Management; d) Discovering and Monitoring; e) Grid Applications Development.
- Each phase has to define ordered EAs for teaching the main training concepts related to the learning modules, and the status of each EA has to be kept (check pointed) at the end of each EA so further EAs are built on top of previous ones.
- The results of each EA will contribute to the creation of a *Target VO* or to its validation.
- All EAs will be assessed through a portfolio where all pieces of evidence (configuration files and source code) generated to accomplish the learning objectives planned in each EA are collected.
- The **Target VO** will be administered by the teacher, but all students will be trained to create and administer VOs, acting as a *VO Admin*.

- Each student will individually represent a different administrative domain acting as a virtual research center or university. Therefore, the student will be the responsible person and the administrator of a simulated administrative domain belonging to the *Target VO*, acting as the *Domain Admin*.

- Each student will manage and configure one computational and storage resource of the administrative domain that represents. Therefore, the student will be the responsible and administrator of the resources, acting as *Resource Admin*.

- Students will belong to the *Target VO* and, therefore, they will be able to develop and run Grid applications using the *Grid Infrastructure* provided by the *Target VO*, acting as a Grid Users.

- The teacher will be able to scale the *Grid Infrastructure* with new simulated administrative domains to include new computational and storage Grid resources to the *Target VO*, and to test the scalability of the Grid applications developed by the *Grid users*.

- Finally, each student will develop a set of advanced Grid applications to validate the *Target VO*, and to test the scalability on top of the virtualised Grid Infrastructure resources provided by the *Target VO*.

Table 1 shows the learning objectives of the stages together with the related EAs planned in the PBL. Figure 1 shows the chronological order and dependencies among all EAS during the five weeks employed to carry out the PBL activities. The next subsections describe, for each EA: a) Scope of the work and the roles assumed by the student; b) portfolio evidences to be assessed by the teacher and required resources; c) a brief explanation about which EAs are new or have been updated to be integrated in the PBL, highlighting the new learning objectives that have been addressed thanks to the new approach. Also, Table 2 shows the teaching methods employed in each EA and a brief description of the contents that are developed therein.

Table 1: List of learning objectives (LO.X) and the EAs of the PBL where they are addressed. In bold the new LO addressed in the new PBL approach.

Stage	Learning Objectives (L.O)	EAs
Grid Security	**LO.1 - Create a VO**	EA1, EA7
	LO.2 - Manage certificates belonging to an Administrative Domain	EA2
	LO.3 - Manage (add/remove) administrative domains in a VO	EA3, EA7
	LO.4 - Request Grid users certificates	EA4
	LO.5 - Add/remove Grid users from administrative domains in the VOs	EA7, EA4
	LO.6 - Request Grid resources certificates	EA5
	LO.7 - Add/remove resources from administrative domains in the VOs	EA5, EA7
	LO.8 - Authorisation management of Grid users to VO resources	EA6, EA7
Grid Job Management	**LO.9 - Configure compute resources using Globus Toolkit**	EA8
	LO.10 - Design and implement basic Grid apps using Globus CLIs	EA9
	LO.11 - Manage the lifecycle of Grid jobs	EA9
Grid Data Management	**LO.12 - Configure data resources using Globus Toolkit**	EA10
	LO.13 - Design and implement basic data Storage Grid apps using Globus CLIs	EA11
	LO.14 - Manage the life-cycle of Data Grid jobs	EA11
Information System	**LO.15 - Configure Globus MDS Service**	EA12
	LO.16 - Publish personalised information of the resource using MDS	EA13
Advanced Grid Apps	LO.17 - Implement Massive Data Processing executions	EA14
	LO.18 - Implement Multi-parametric executions	EA15
	LO.19 - Implement Loosely Coupled Parallel executions	EA16

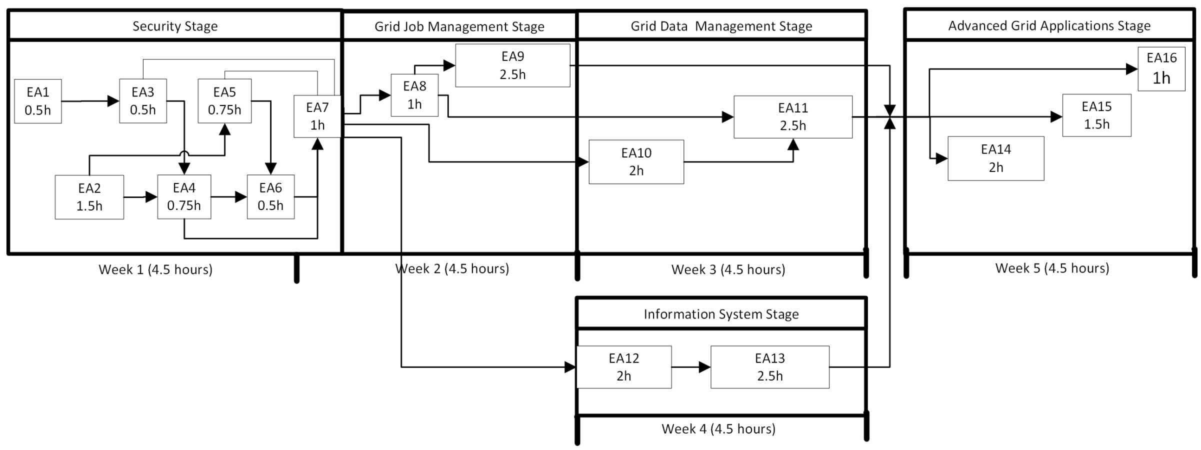

Figure 1: Chronological order of all AE and dependencies of the stages defined in the PBL for every week.

4.1. Grid Security Stage

This stage is composed of 7 EAs. The objective of this stage is to create a secure *Target VO*, comprising a set of fictional institutions (representing different administrative domains). Each administrative domain will be managed by a student and it will have a *Grid user* belonging to the *Target VO*.

4.1.1. VO Creation (EA1)

In this EA, each student (*VO Admin*) learns how to manage the Virtual Organisation Membership Service (VOMS)[4] for creating a VO.

As evidence to be evaluated in the portfolio, each student creates his/her own VO through a *VOMS node*.

The EA1 is a new activity designed at GCCC which allows addressing a new Learning Objective in the course (L01, see Table 1). To carry out this EA, we implemented the GaaS cloud-based tool which will be presented in the next section. GaaS deploys a *VOMS node* on the cloud so the teacher and the student

[4]Virtual Organization Membership Service: `https://italiangrid.github.io/voms/`

Table 2: Activities and employed teaching method. MC (Master Class), Demo (Demonstration guided by Professor).

EA	Description of Method
EA1	MC - Concept of VO. Demo - How to manage VOs using the VOMs service.
EA2	MC - Fundamentals of asymmetric cryptography. Demo - How to manage certificates using an OpenCA PKI.
EA3	Demo - How to add an administrative domain in a VO.
EA4	MC - Basics of Grid Users certificates. Demo - How to request a Grid User certificate and membership to a VO.
EA5	MC - Basics of Grid Resource certificates. Demo - How to request a Resource User certificate.
EA6	Demo - How to configure a resource to authorise Grid users.
EA7	Demo - How to create a VO and to integrate new Grid users and Resources.
EA8	MC - Basics of GRAM service architecture and its functionalities. Demo - How to configure the GRAM service.
EA9	MC - Basics of Grid Application architecture. Demo - How to build a basic Grid application and manage its lifecycle using Globus CLIs.
EA10	MC - Basics of GASS and GridFTP services architecture and their functionalities. Demo - How to configure the GASS and GridFTP services.
EA11	MC - Basics of Data Storage Grid applications architecture. Demo - How to build a Storage Grid application and manage its lifecycle using Globus CLIs.
EA12	MC - Basics of MDS service architecture and its functionalities. Demo - How configure and manage the MDS service.
EA13	Demo - How to personalise information in the MDS.
EA14	MC - Basics of Massive data processing Grid application architecture. Demo - How to build a Massive data processing Grid application.
EA15	MC - Basics of Multi-parametric execution Grid application architecture. Demo - How to build a Multi-parametric execution Grid application.
EA16	MC - Basics of Loosely Coupled Parallel Grid application architecture. Demo - How to build a Loosely Coupled Parallel Grid application.

can access and administer them without limitations. Notice that the real VOs used provided a VOMS service that is managed by an external institution, and thus the students could not create new VOs.

4.1.2. PKI Management (EA2)

In this EA, the student acquires the *Domain Admin role* and learns how to manage certificates through a Public Key Infrastructure (PKI) belonging to a simulated Administrative Domain.

As portfolio evidence, students install and configure their own OpenCA PKI[5] through a *CA node*.

The EA2 is a new activity designed at GCCC which tackles L0.2 (see Table 1) by using GaaS to deploy a *CA node* per student on the cloud. The teacher and the students can access the *CA Nodes* without limitations. The course used real PKIs managed by external institutions, and thus they could not be used to sign or revoke new certificates by the students.

4.1.3. VO Management (EA3)

In this EA, the students (*VO admin*) learn how to integrate different administrative domains in a VO through the configuration of the VOMS Service.

As portfolio evidence, each student inserts their own Administrative Domain (EA2) in their VO (EA1) through the *VOMS node*.

This is a new activity which allows addressing L0.3 (see Table 1) because the real VO used up to this moment could not be managed by the students.

4.1.4. Grid User Creation (EA4)

In this EA, the students learn to create Grid User certificates in five steps. First, the students (*Grid User*) simulate a user who belongs to a given Administrative Domain. Second, each student requests a Grid User certificate of his/her administrative domain. Third, the students (*Domain Admin*) sign the

[5]OpenCA PKI: `https://www.openca.org/projects/openca/`

request certificates using the PKI infrastructure from their Administrative Domain. Fourth, they (*Grid User*) send a membership request to a VO. Finally, students (*VO Admin* accept the membership of the Grid User to the VO through the VOMS service.

As portfolio evidence, each student creates a request of a Grid User Certificate through the *Globus node*, sign it through the *CA node* and registers the user to the VO through the *VOMS node*.

The EA4 is a partially new activity in GCCC. The last three steps required the use of GaaS. The first step is an old activity performed at GCCC that has been updated and integrated to the PBL target. Thus, the last three steps allows addressing LO.5.

The EA4 has been redesigned to adapt it as a stage in the PBL. In the former courses of GCCC, this activity was carried out partially due to the lack of resources to implement EA1 and EA2. Only the request of a Grid User certificate (L0.4) to the existing CAs in the real VO was possible.

4.1.5. Grid Resource Certification (EA5)

In this EA, the students learn to create Grid Resource Certificates in three steps. First, the students (*Resource Admin*) simulate a site administrator who manages a resource in a given administrative domain. Second, each student requests a Resource Certificate of his/her administrative domain. Third, the students (*Domain Admin*) sign the request certificates using the PKI infrastructure of their administrative domain. Finally, they (*Admin role*) configure the certificates in the Grid Resource.

As portfolio evidence, each student creates a request of a Resource Certificate though the *Globus node* and sign it through the *CA node*.

The EA5 is a new activity scheduled in GCCC that also uses the GaaS. EA4 and EA5 have been redesigned to adapt them as stages in the PBL and allow addressing LO.7. In the previous approach only requesting a certificate (LO.6) as a Resource Admin to the existing CAs could be addressed.

4.1.6. Grid user Authorization (EA6)

In this EA, each student (*Resource Admin*) learns how to configure the resource to accept the access from all the Grid users belonging to the VO.

As portfolio evidence, students configure their resource to authorise their Grid User through the *Globus Node*.

The EA6 is a new activity programmed at GCCC which allows addressing LO.8. This EA required the GaaS to manage the Globus Nodes. Previously, it was impossible to access the resources because they were managed by external institutions.

4.1.7. Target VO Creation (EA7)

In this EA, the teacher first creates the *Target VO* to integrate all the administrative domains, Grid users and Grid resources created in EA3 and EA4. Next, each student (*Grid User role*) incorporates his/her Grid User to the VO. Finally, the students (*Resource Admin*) incorporate their resources.

As portfolio evidence, the target VO has been built incorporating all Grid Users and resources through the *VOMS Service*. The target VO will be used in the next stages of the PBL.

The EA7 is a new activity programmed at GCCC because the real VO used up to this moment could not be created by the teacher. This EA also address the LO.1, LO.3, LO.5, LO.7 and LO.8.

4.2. Grid Job Management Stage

This stage is composed of two EAs and the goal is to configure a set of Grid resources belonging to the secure *Target VO*, one per each fictional administrative domain that was created in the previous stage, to enable and validate Grid computing resources in the VO.

4.2.1. Computing Grid Resources Configuration (EA8)

In this EA, each student (*Resource Admin*) learns how to configure the Globus Resource Allocation Manager service (GRAM) [23] in the Grid Resources for submitting computing jobs.

As portfolio evidence, students configure their GRAM service through their *Globus Node*.

This is a new activity programmed at GCCC which allows addressing LO.9. In the previous scenario, the resources used to be provided and managed by external institutions, and thus they could not be configured by the students.

4.2.2. Basic Grid Applications Development (EA9)

In this EA, each student (*Grid User*) learns how to design, deploy and manage the lifecycle of a set of Grid applications using the CLIs offered by the Globus Toolkit.

As portfolio evidence, students create and test a set of basic Grid applications through their *Globus Node*.

This is an existing activity programmed at GCCC. Previously, the same activity was executed but using the Grid Infrastructure provided by the Virtual Organisations of the ES-NGI or EGI.

4.3. Grid Data Management Stage

This stage is composed of 2 EAs. The objective is to configure a set of Grid resources belonging to the secure *Target VO*, one per fictional administrative domain that they were created in the previous stage, to enable and validate Grid Data resources in the VO.

4.3.1. Data Grid Resources Configuration (EA10)

In this EA each student (*Admin role*) learns how to configure the Global Access to Secondary Storage (GASS) [24] and GridFTP [25] servers.

As portfolio evidence, student configure their GASS and GridFTP services through their *Globus Node*.

This is a new activity programmed at GCCC which allows addressing LO.12. Previously, the resources used to be provided and managed by external institutions, and thus they could not be configured by the students.

4.3.2. Basic Data Storage Grid Applications Development (EA11)

In this EA, each student (*Grid User*) learns how to design, deploy and manage the lifecycle of a set of Data Storage Grid applications using the CLIs provided by the Globus Toolkit.

As portfolio evidence, students create and test a set of storage Grid applications through their *Globus Node*.

This is is an existing activity designed at GCCC. Previously, the same activity was executed but using the Grid Infrastructure provided by the Virtual Organisations of the ES-NGI or EGI.

4.4. Information System Stage

This stage is composed of 2 EAs. The objective is to configure a Monitoring and Discovering System to manage the resources of the secure *Target VO*.

4.4.1. Monitoring System Configuration (EA12)

In this EA, each student (*Resource Admin role*) learns how to configure the Monitoring and Discovering Service (MDS) [26].

As portfolio evidence, students configure their MDS service through their *Globus Node*. This is a new activity designed at GCCC which allows addressing LO.15. The MDS used previously were managed by external institutions.

4.4.2. Monitoring System Configuration (EA13)

In this EA, each student (*Resource Admin*) learns how to modify the MDS service to publish specific and personalised information of the resource.

As portfolio evidence, students personalise the information in their MDS service through their *Globus Node*.

This is a new activity programmed at GCCC which allows addressing LO.16.

4.5. Advanced Grid Applications Deployment Stage

This part is composed of 3 EAs (EA14, EA15 and EA16). The objective is to design and implement Advanced Grid applications using all Grid resources offered by the *Target VO*.

In these EAs, the students (*Grid user*) learn how to design, deploy and manage advanced Grid applications based on Massive Data Processing, Multi-parametric execution and Loosely Coupled Parallel Application respectively.

As portfolio evidence in each EA, each student creates and tests a set of Grid applications through his/her *Globus Node*.

These are existing activities in GCCC as EA9 and EA11, and they were previously executed on production Grid Infrastructures.

5. Grid as a Service cloud-based tool

This section describes the Grid as a Service (GaaS) cloud-based tool that has been designed and built to drive the PBL exposed in section 4. Four steps are used to develop these types of cloud services. First, a timing analysis about when the identified computational resources must be created, stopped or removed is carried out. Second, the software, hardware and configuration of these resources is defined. Third, a list of the functional and non-functional requirements for the cloud-based tool is compiled. Finally, based on the requirements an architecture is defined and implemented.

The following subsections describe the previously discussed steps applied to the design and implementation of GaaS, considering the educational framework.

5.1. Timing Requirements

Table 3 shows the computational resources and their software, hardware and configuration to perform all EAs defined at the PBL.

In the context of the EAs defined in the PBL, the timing requirements of these computational resources are shown in Figure 2. We can see in this figure when each computational resource should be deployed, paused or terminated, or even when the state of a given node has to be maintained (checkpoint).

5.2. Functional and Non-Functional Requirements

Table 4 shows the functional requirements for the educational framework defined.

Table 3: Nodes required to perform the EAs defined at the PBL. Software and hardware requirements together with the configuration needed.

Node	O.S	N	Hardware	Sofware	Configuration
Globus	Scientific Linux	1 per student	4GB, x86-64, 20GB HD	Globus Toolkit 6	Accounts to develop grid applications and manage the resource. Access via SSH with a password.
Advanced Globus	Scientific Linux	Indeter.	4GB, x86-64, 20GB HD	Globus Toolkit 6	GRAM, GASS, GridFTP, MDS and certificates installed and configured.
VOMS	Scientific Linux	1	4GB, x86-64, 10GB HD	VOMS Server	A VOMS admin user to access via SSH with a password and a web browser.
CA	Ubuntu	2 per student	1GB, x86-64, 10GB HD	OpenCA 1.4	A CA admin user to access via Web browser through user and password.

Table 4: Functional requirements of GaaS Service and the EA associated.

Req.ID	Description	EAs
R.1	It must deploy a VOMS node (Deploy VOMS) that will be shared by all the students.	EA1
R.2	It must pause the VOMS node	EA2, EA5, EA6, EA8, EA10, EA12, EA13
R.3	It must restart the VOMS node	EA3, EA7, EA9, EA11, EA14
R.4	It must save or restore (checkpoint) the VOMS Node	EA1, EA3, EA4, EA7
R.5	It must deploy one CA node per student (Deploy All CAs)	EA2
R.6	It must pause CA nodes	EA3
R.7	It must restart all CA nodes	EA4
R.8	It must save or restore (checkpoint) the CA Nodes	EA2, EA4, EA5
R.9	It must deploy one Globus node per student (Deploy All Globus)	EA4
R.10	It must save or restore (checkpoint) the Globus Nodes	EA4, EA5, EA6, EA8, EA13
R.11	It must deploy a certain number of Advanced Globus Nodes (Deploy All Advanced Globus) for the scalability study of Grid applications	EA14
R.12	It must terminate any nodes that are no longer required in the PBL	EA5, EA16
R.13	It must allow the students access to any node via internet through SSH for using the Command Lines Interfaces (CLIs)	All E.As
R.14	It must allow the students to access CA and VOMS nodes via internet through HTTPs for managing web interfaces	EA1, EA2, EA3, EA4, EA5, EA7, EA9, EA11, EA14, EA15, EA16

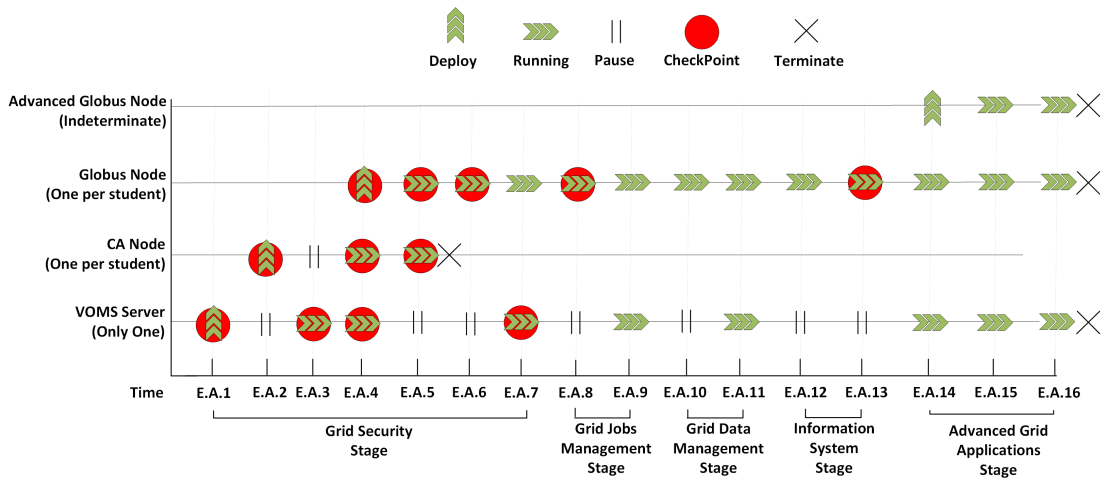

Figure 2: Sequence of the EAs defined in the PBL.

It is important to highlight R.8 and R.10, since replacements for malfunctioning nodes can be deployed on-demand and the students can continue performing the subsequent E.As restoring the status of the previous EA. This is a frequent issue due to severe misconfiguration steps from the students in which a fresh start with a new node is recommended.

The non-functional requirements for the GaaS are described in Table 5:

5.3. GaaS Architecture

Based on the requirements presented above, this section describes the architecture of the GaaS to deploy the virtualised Grid infrastructures on top of Cloud providers and driven the PBL presented in section 4.

The platform to develop the GaaS is ODISEA [27, 28]. ODISEA enables deploying virtual infrastructures through recipes that describe their hardware, software and configuration needed, using a declarative language named RADL (Resource Application & Description Language) [27]. It also supports the TOSCA Simple Profile in YAML Version 1.1 standard [28]. The core component of ODISEA is the Infrastructure Manager [27] (IM) which can deploy an

Table 5: Non-Functional requirements of the GaaS Service.

Req.ID	Description
R.1	It must provide and configure the underlying computational infrastructure (Virtual Machines - VMs) from the Cloud provider to perform the EAs defined at PBL.
R.2	It must offer a friendly user interface, showing information about the deployed nodes and their status.
R.3	It must deploy the nodes in an agnostic and transparent way in different cloud providers (public or private).
R.4	It must reduce the deployment time compared to a manual deployment.
R.5	It must enable all nodes to communicate through a private network to isolate the traffic.

infrastructure described in RADL in different cloud providers (such as AWS, Microsoft Azure, OpenNebula and OpenStack). The IM has been subsequently improved across several European research projects, such as INDIGO-DataCloud[6] and EOSC-HUB[7] and it has been integrated in the Federated Cloud AppDB VMOps Dashboard[8] as the cloud orchestrator in the large-scale Federated European Cloud infrastructure. The following subsections describe the components and features of each node along with the interactions through the developed functionalities of the GaaS.

5.3.1. Grid Project Based Learning Front-End Node

The GaaS architecture defines a PBL Front-End (PBL-FE) node to manage the deployment, termination, stop, restart, save and restore of the nodes of the

[6]INDIGO-DataCloud: `http://www.indigo-datacloud.eu`

[7]EOSC-HUB: `https://www.eosc-hub.eu`

[8]Federated Cloud AppDB VMOps Dashboard: `https://wiki.egi.eu/wiki/FederatedCloudAppDBVMOpsDashboard`

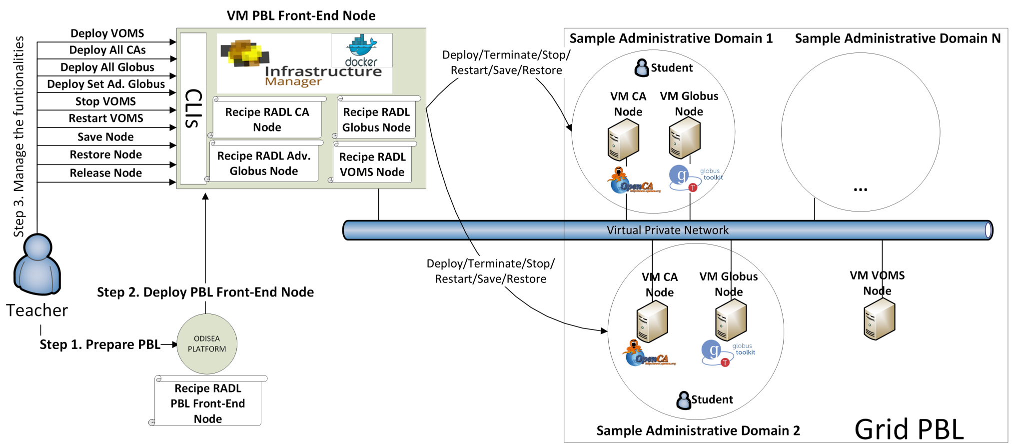

Figure 3: Architecture of the GaaS to deliver Grid Infrastructures as a Service for education.

infrastructure (*VOMS Node*, *CA Nodes*, *Globus Nodes* and *Advanced Globus Nodes*) associated to the EAs of the PBL. The details of the infrastructure for this task (see step 1 in Figure 3) are defined by a RADL recipe (see listing code 1) that it is used to automatically deploy the *PBL-FE* using the ODISEA platform in an on-premises or public cloud provider (see step 2 in Figure 3).

The *PBL-FE* provides the teacher with a set of CLIs that enables the functionalities described in the table 4 (see step 3 in Figure 3). The CLIs directly interact with the IM hosted in the PBL-FE through a Docker container (see Figure 3). In addition, the *PBL-FE* has the required RADL recipes for deploying and releasing the *VOMS Node*, *CA Nodes*, *Globus Nodes* and *Advance Globus Nodes* needed for implementing the PBL. Also, IM provides the functionalities to stop nodes or copy images (checkpoint) of the running nodes and clone them.

The *PBL-FE* also acts as a bridge that provides a communication channel for all deployed nodes through a virtual private network (see Figure 3). The communication channels among the nodes are implemented using rsync[9]

[9]`https://rsync.samba.org/`

to synchronise the CA certificates from *PBL-FE Node* to the *Globus Nodes* and *Advanced Globus Nodes*. It also uses the following 0MQ sockets[10], specifically, the socket channels are the following: :

- CA Socket REQ: One socket is connected to each *CA Node* and it is responsible for requesting the required certificates to configure the *Advanced Grid Nodes*.
- Adv. Globus Socket REQ: One socket is connected to each *Advanced Grid Node*, which is in charge of notifying that the credentials are already transferred and that everything is ready to start the Globus services, leaving the nodes ready to be used.

Listing 1: Fragment of the RADL document for the dynamic deployment of the PBL-FE node.

```
network net (outbound = 'yes' ...)
system nodeFE (
    cpu.arch = 'x86_64' and cpu.count >= 1 and
    memory.size >= 512m and disk.0.os.name = 'linux' and
    disk.0.os.flavour = 'ubuntu' and ...
)
configure configureFE (
@begin
    - tasks:
            - name: Installing required software
              apt: state=latest update_cache=yes ...
            - rsync
    - name: Installing required python modules
        pip: state=latest ...
        - pip, setuptools, zmq, im_client
@end
```

[10] 0MQ: http://zeromq.org/

```
)
...
@end
)
contextualize (
    system nodeFE configure configureFE
    system nodeFE configure dockerizeIM
)
deploy nodeFE 1
```

5.3.2. VOMS Node

The *VOMS node* is used for managing the groups, roles and Grid users of a given VO. The *VOMS Node* has the Virtual Organization Membership Service (VOMS)[11] installed and configured. Figure 4 describes the interactions between the *VOMS Node* and the *PBL-FE Node* to perform the functionalities defined in the requirements (see section 5.2). Those functionalities are:

- **Deploy VOMS.** Only one *VOMS Node* is deployed for all students to manage VOs. The VOMS certificate and CA certificate used for signing the VOMS Certificate are synchronised using rsync in the *PBL-FE Node*.
- **Stop VOMS.** The Virtual Machines (*VOMS node*) is stopped.
- **Restart VOMS.** The Virtual Machine (*VOMS Node*) is restarted.
- **Save VOMS.** The Virtual Machine image of the *VOMS Node* is saved as a checkpoint.
- **Restore VOMS.** The *VOMS Node* is restored from the previously saved checkpoint.

[11]Virtual Organization Membership Service: `https://italiangrid.github.io/voms`

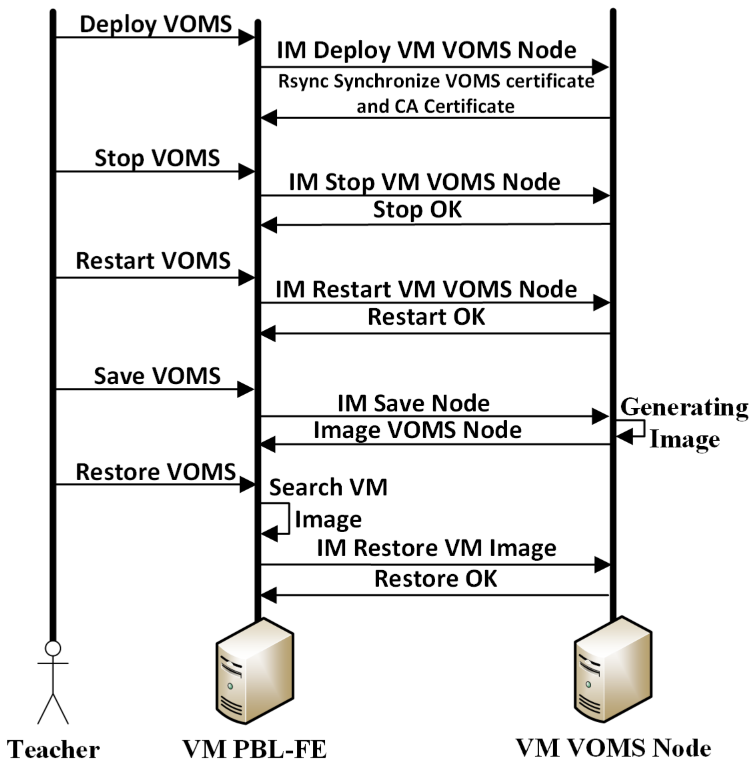

Figure 4: Workflow associated with the *VOMS Node* functionalities.

5.3.3. Certificate Authority Node (CA Node)

The *CA Node* is a Certification Authority (CA) for an administrative domain that belongs to a VO. The CA Node has the OpenCA[12] installed and configured. It is used to sign the certificates of the Grid users and Grid resources of each domain. This node needs the port 80 to be accessible from outside to enable the interaction with OpenCA using the web interface.

In addition, each *CA Node* synchronises the CA certificate with the *PBL-FE Node* and has a socket (PBL-FE Socket REP) that will receive a request from *PBL-FE* to execute a script that automatically signs resource certificates request for a given *Advanced Grid Nodes* and return the signed certificates and the confirmation that the operation has been performed successfully.

Figure 5 describes the interactions among the *CA Nodes* and the other nodes required to perform the functionalities (see section 5.2). These functions are:

- **Deploy all CAs.** One *CA Node* is deployed per student to represent different administrative domains.
- **Save CAs.** Virtual Machine images of active *CA Nodes* are saved as a checkpoint.
- **Restore CA.** A specific *CA Node* is restored from the previously saved checkpoint.

5.3.4. Globus Node

The *Globus Node* acts as a computation and storage resource in the different administrative domains belonging to a VO. The *Globus Node* has the Globus Toolkit installed and will be used in the different EAs to configure Grid services. When the *Globus Node* is deployed, the *PBL-FE* is in charge of uploading all CA certificates registered and synchronises them with all the *Globus Nodes* to enable them to: a) request a certificate signed from any CA; b) accept user connections from any simulated administrative domain.

[12]Open CA: `https://www.openca.org/`

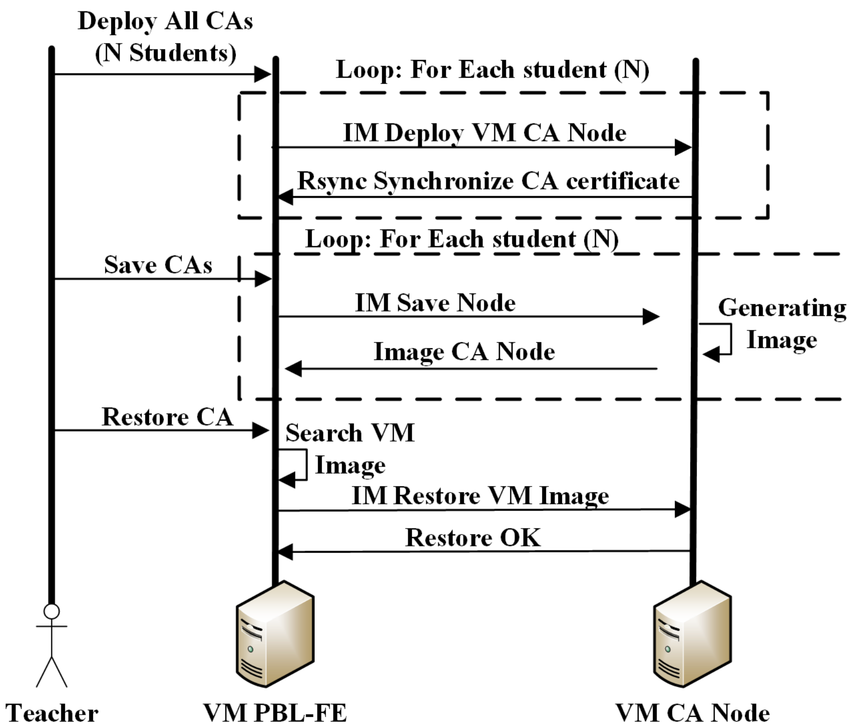

Figure 5: Workflow associated with the *CA Node* functionalities.

Figure 6 defines the interactions among the *Globus Nodes* and the other nodes required to perform the functionalities (see section 5.2) requested. These functionalities are:

- **Deploy all Globus.** *Globus Nodes* are deployed for each student in their own administrative domains.
- **Save Globus.** Virtual Machine images of active *Globus Nodes* are saved as a checkpoint.
- **Restore Globus.** A specific *Globus Node* is restored from the previous saved checkpoint.

5.3.5. Advanced Globus Node

The *Advanced Globus Nodes* are used to scale up and down the Grid infrastructure. The difference with the *Globus Node* is that it is fully configured and ready to be used in the Grid. For this, it includes certificates signed by a CA and it is configured for communication through the virtual private network. Also, the node has a socket (*Advanced Grid Node* Socket REP) that allows the *Advanced Grid Nodes* to automatically start up the Globus services when all certificates have been synchronised and confirm that the required configurations have been automatically performed.

Figure 7 describes the interactions among the *Advanced Globus Nodes* and the other nodes required to perform the functionality (see section 5.2) defined. The required function is **Deploy VM Advanced Grid nodes** where a number of *Advanced Globus Nodes* are deployed.

6. Results and Discussion

First, an assessment of the student's opinion about the PBL performed using the GaaS is presented. Next, two statistical analyses have been carried out to detect imbalances of complexities between PBL stages or EA in the same stage. After that, the cloud-based tool is evaluated in two aspects: the analysis of

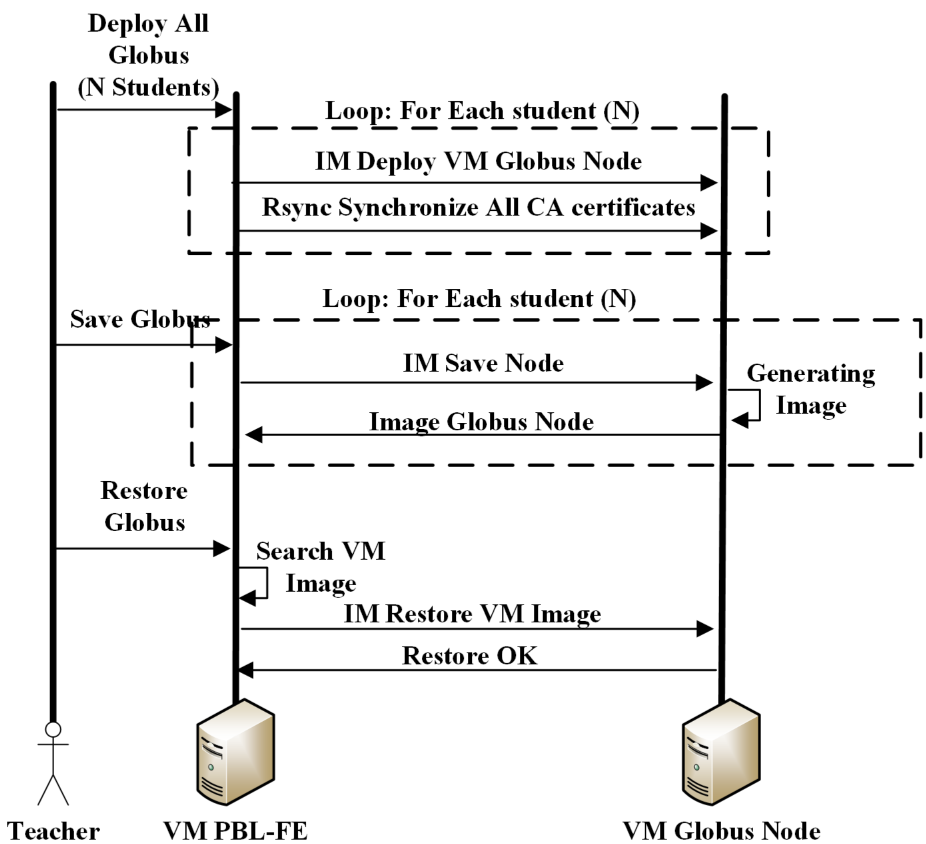

Figure 6: Workflow associated with the *Globus Node* functionalities.

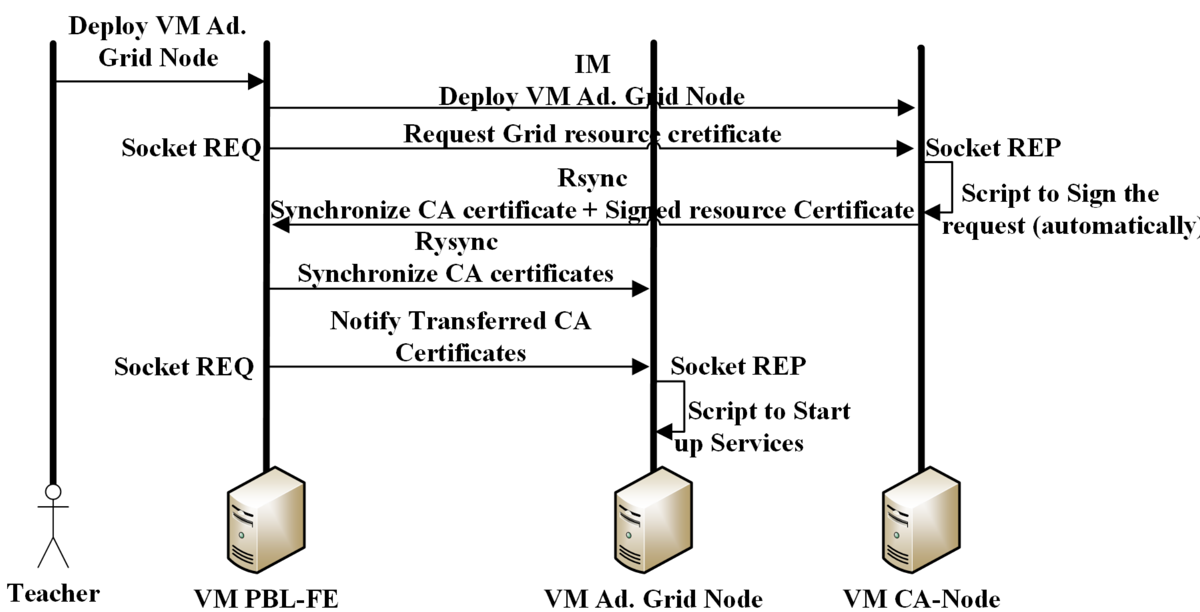

Figure 7: Workflow associated with the *Advanced Globus Node* functionalities.

Table 6: Questionnaire employed to gather student feedback and number of student answers. (TD) Totally Disagree, (D) Disagree, (F) Fair, (A) Agree, (TA) Totally Agree.

ID	Question	TD	D	F	A	TA
Q1	The methodology employed in CCGC allowed me to reach the learning goals from the point of view of a Grid user for the development of Grid applications.	0	0	1	11	24
Q2	The methodology employed in CCGC allowed me to reach the learning goals from the point of view of administrator roles (Domain Admin, VO Admin and Resource Admin) of a Grid infrastructure.	0	0	6	15	15
Q3	The collaborative creation of a Grid infrastructure among the students is a good educational methodology to reach the learning goals.	0	0	2	4	30
Q4	The Virtual Machines were always available to carry out the practice sessions.	0	0	0	0	36

the deployment times of the main nodes on an on-premises Cloud and a cost analysis on the impact of moving the required nodes to a public cloud provider.

6.1. Questionnaire Assessment

As stated earlier, the implemented GaaS has been incorporated in the Master's Degree in Cloud Computing and High Performance Computing (MUCNAP) at the Universitat Politècnica de València (UPV), in Spain, where the GaaS has been used in the subject Grid Computing and Cloud Concepts (GCCC). A total of 36 students were evaluated using the questionnaire shown in Table 6. In the questionnaire, the students express their opinion on the new methodology and also if they consider that the formative objectives have been accomplished from the Grid user and administrator point of view. The answers are measured using a Likert scale from 0 (totally disagree) to 5 (totally agree).

Figure 8 shows the results of the questionnaire from Table 3. Concerning Q1, 97,2% of the students "Agree" or "Totally Agree" with the statement and, therefore, they think that the learning goals concerning the development of Grid applications have been reached. This is mainly due to the adoption of highly mature activities that have been employed in the last years in several VOs (e.g. tut.vo.ibergrid.eu) and laboratories (e.g. Gilda). Therefore, it is clear that the PBL conducted through the GaaS developed in this work facilitates carrying out those activities.

Concerning Q2, 83.3% of students "Agree" or "Totally Agree" with the statement and, therefore, they consider to have reached the learning goals concerning the administration of a Grid infrastructure. Despite being a good score, the difference with respect to the previous question could be caused by the lower maturity of the Grid administration activities. It is expected that this value increases in the upcoming academic courses.

Concerning Q3, 94,4% of students found the use of GaaS for the PBL is a good approach to reach the learning goals. Therefore, combining methodologies based on PBL with Cloud computing stands out as an appropriate combination to provide the resources required for its execution.

Finally, Q4 points out one of the main advantages of using Cloud Computing, as described in the work by González-Martínez et al. [29], which is high availability and the ubiquitous access to resources. Indeed, 100% of the students had available the resources anytime in order to carry out the EAs from the PBL.

6.2. Imbalances of Complexities

The 20% of the GCCC final grade is done through the assessment of all the EAs through a portfolio. The portfolio is made up of pieces of evidence that the student generates in the nodes deployed with the GaaS required for each activity. These pieces of evidence can be a configuration file, a log, X.509 certificates and source codes of Grid applications, among others. The teacher evaluates (using a scale from 0 to 10) all the EAs accessing the nodes through GaaS. In this section, first a comparison is made of the degree of complexity

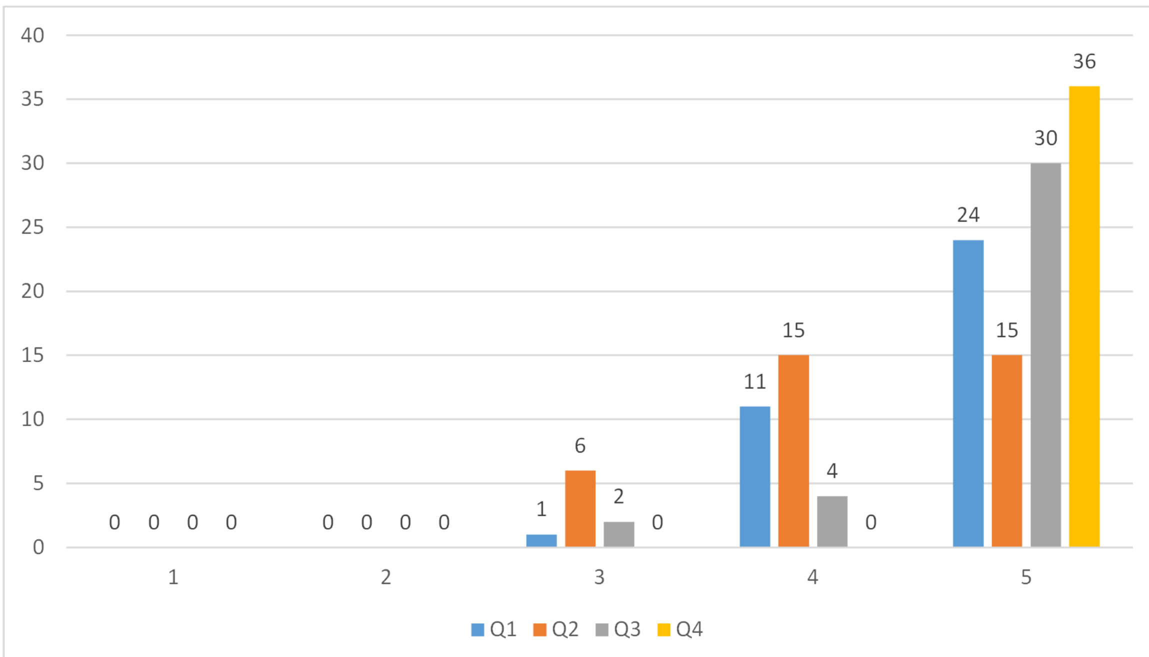

Figure 8: Results from the questionnaire.

Table 7: Descriptive statistics about the final marks of the stages planned at designed PBL.

	Grid Security (Stage 1)	Job Mng. (Stage 2)	Data Mng. (Stage 3)	Inf. system (Stage 4)	Adv. App. (Stage 5)
Avd	9.27	8.51	8.44	9.08	8.86
Std. Desv.	1.34	1.98	2.00	1.68	1.51
K-S	0.000	0.000	0.000	0.000	0.000

between the five phases defined in the PBL. The objective of this first study is to see if there are imbalances between the phases in terms of complexity of development by the student. Next, a second study is carried out on the differences between the degree of complexity between the EAs of the same phase, with the aim of balancing the EAs of the same phase. In both studies, to measure the degree of complexity, the portfolio evaluations made by the teacher in each EA based on the evidence generated by the 36 students have been taken as a quantitative measure. For the comparison in both studies, Friedman's two-way non-parametric statistical test has been used for related samples [30], since none of the distributions is normal.

Table 7 shows the statistical descriptive analysis of each of the phases implemented in the PBL, where the average of the set of final marks of the EAs of each phase is shown. Also in Table 7 we see that the distributions of each phase are not normal, since the Kolmogorov-Smirnov test results in p (K-S) < 0.05, rejecting the null hypothesis. We observe that phase 2 and 3 have the minimum averages and that it could assume that their degree of difficulty is greater. In Table 8 we see that effectively phases 2 and 3 have significant differences with phase 1, but not with the rest of the phases. Phase 1 is very focused on concepts related to security, this part being more familiar to all students, and in which many of them already come with background knowledge. However, the rest of the phases focus more on the development of Grid applications and

Table 8: Friedman Test of PBL Stages Final Marks.

Frd (p)	Stages	Frd (p)	E.As	Frd (p)	EAs	Frd (p)	EAs	Frd (p)
0.000	1&2	**0.015**	2&3	1.000	3&4	0.081	4&5	1.000
	1&3	**0.004**	2&4	0.230	3&5	0.442		
	1&4	1.000	2&5	1.000				
	1&5	1.000						

configuration of Grid resources, this part being very new for everyone, so it can lead to the completion of phase two and three which significantly more complex for the student. However, once the concepts of these two phases have been assumed, the realization of the rest is more affordable, since they have already consolidated basic concepts of Grid Computing. Therefore, it would be convenient to balance these phases and make the transition from phase 1 to phase 2 smoother, reallocating more time to phase two to the detriment of phase 1. In this first analysis, we can say that the phases are reasonably balanced in degree of complexity.

For the second analysis, we observe in Table 9 the descriptive statistics of the 16 EAs of the PBL. As in the previous study, we can see that none of the distributions is normal, given that p (K-S) < 0.05, and therefore we have applied Friedman's two-way test. Table 10 shows that in phase 1 the null hypothesis p (Friedman) < 0.05 is rejected, so there are significant differences between the 7 distributions that make up the first phase. If we look at the detail, we see that the EA2 is the only EA that differs from the rest of the EAs of phase 1, so its degree of difficulty is greater than the rest of the class activities in this first phase. In a part of EA2, asymmetric cryptography is explained, and it will probably take more time to consolidate the theoretical concepts related to this matter given its complexity. The rest of the activities are more focused on the use of functionalities of different services, so the realization of these is easier for the students.

Table 9: Descriptive statistics about the final marks of the Educational Activities (EAs) planned at designed PBL stages.

EA	1	2	3	4	5	6	7	8	9	10
Avd.	9.47	8.36	9.44	9.50	9.31	9.31	9.50	8.58	8.44	8.53
Std. Desv	1.23	2.38	1.30	1.16	1.35	1.45	1.28	2.09	1.96	2.04
K-S	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
EA	11	12	13	14	15	16				
Avd.	8.36	8.97	9.19	8.64	8.80	9.14				
Std. Desv	2.11	1.90	1.47	1.66	1.58	1.38				
K-S	0.00	0.00	0.00	0.00	0.00	0.00				

In Table 10, we also observe significant differences between the defined EAs in phase 4 and phase 5. In phase 4, the EA12 focuses on an in-depth description of the architecture and management of LDAP services, which is a complex part for the student, while the EA13 is a very guided practical part that is easier to solve by the student. The imbalance in these two tasks could be addressed by cutting back some of the discussion regarding LDAP and only focusing on the MDS service capabilities. Regarding phase 5, we observe that there are differences between the EAs, but in a progressive way as we can see in the average values of EA14, EA15 and EA16 in Table 9. These three EAs have many similar concepts, so it will be reasonable that the students improve the understanding and performance as the phases are completed.

6.3. Deployment Time

Average deployment time (average of three executions) have been measured for the node types (*PBL-FE*, *CA Nodes*, *Globus Nodes* and *Advanced Globus Nodes*). The deployment has been performed in an on-premises cloud with a total of 128 cores and 352 GB of RAM managed by OpenNebula 4.2. We differ-

Table 10: Friedman Test of Educational Activities per Stages.

Stage	Frd (p)	EAs	Frd (p)	E.As	Frd (p)	EAs	Frd (p)	EAs	Frd (p)	EAs	Frd (p)
1	0.000	1&2	**0.012**	2&3	**0.016**	3&4	0.785	4&5	0.530	5&6	0.913
		1&3	0.913	2&4	**0.008**	3&5	0.723	4&6	0.461	5&7	0.604
		1&4	0.870	2&5	**0.041**	3&6	0.643	4&7	0.913		
		1&5	0.643	2&6	0.053	3&7	0.807				
		1&6	0.567	2&7	**0.010**						
		1&7	0.956								
2	0.403										
3	0.429										
4	**0.033**	12&13	**0.033**								
5	**0.000**	14&15	0.316	15&16	0.178						
		15&16	**0.012**								

entiate between two metrics: a) Deployment: time spent from the submission of a VM until it is ready for the configuration to start; b) Contextualization: time required for the installation and configuration of the required packages.

Figure 9 shows that the deployment time is similar for all the nodes. However, differences arise in the configuration phase due to the different software packages required and the different complexity of the configuration tasks for each node type. The *CA Nodes* involve the longest time mainly due to the configuration tasks, while the rest of nodes (*Globus Nodes*, *Advanced Globus Nodes* and *PBL-FE Node*) have similar configuration times since only a few packages have to be installed and configured.

To perform the PBL, the first operation required is to deploy the *PBL-FE* (around 7.5 minutes) and then interact with this node to deploy the nodes required in each EA The deployment of *CA Nodes* (EA2) and *Globus Nodes* (EA3) is carried out in parallel, and thus requires approximately 20 minutes for *CA Nodes* and 5 minutes for *Globus Nodes*. The installation of the *CA*

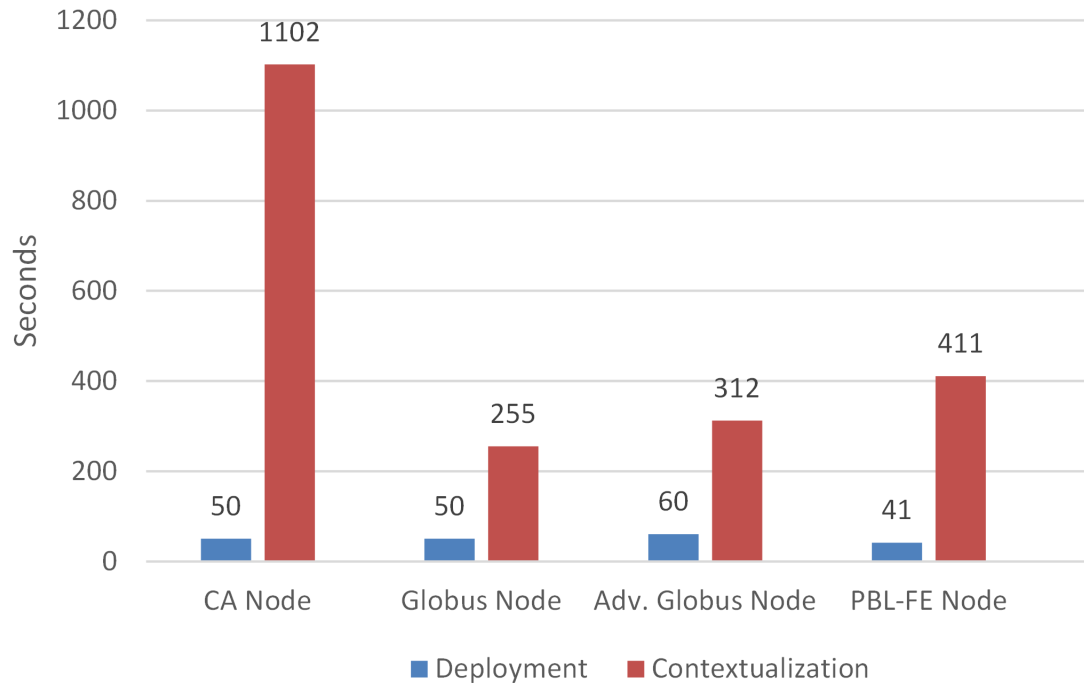

Figure 9: Times per Node.

Nodes requires the interaction with the students who introduce the data of the institution that represents their simulated administrative domain (which typically take around 4 minutes).

It is important to point out that the whole environment to perform the EA2 (the operation requiring the longest time) can be automatically deployed in approximately half an hour. Then, in just five minutes, the *Globus Nodes* (EA3) also can be deployed. Furthermore, new *Globus Nodes* can be deployed if required (Checkpoint) due to misconfigurations from the students. Furthermore, the teacher is able to automatically deploy the required infrastructure and scale it depending on the requirements, such as the number of students. The usage of the ODISEA platform allows to deploy the infrastructure on both public and on-premises clouds. Therefore, if the number of students is too high that exceeds the capacity of the on-premises Cloud, it is possible to outsource the infrastructure in a public Cloud provider. The economical impact of moving the GaaS to a public provider, considering a pay-per-use model, is analysed in section 6.4.

Also, this approach provides deterministic and reproducible computational learning environments that can be reused across multiple subjects. This reduces the workload of the instructor who does not have to dedicate time to configure the computational resources. The ability to provide automated configuration for these type of learning environments paves the way for a better rationalisation of computational resources, where educational institutions can provide their own on-premises Cloud infrastructures to support the requirements of computational subjects in cases where public Clouds cannot be used due to cost constraints.

6.4. Cost Analysis

Table 11 shows the cost of deploying this computational infrastructure in a public cloud provider. The table strictly shows the cost that results from the computational resources used in the 4 hours and 30 minutes face-to-face sessions distributed in six different weeks. We used the resource prices of Amazon

Web Services (AWS)[13], as it is the leading public Cloud provider. The cost incurred in each session depends on the number of nodes and the time they are running or stopped. A stopped node generates a marginal cost due to data storage that can be neglected. The costs have been obtained using the AWS Pricing Calculator assuming resources provisioned in the N. Virginia (us-east-1) region. The following instance types have been selected for the different types of nodes (shown in Table 11), depending on the amount of computing and memory resources, and trying to minimise the cost: t2.medium for the *Globus, Advanced Globus and VOMS nodes* and t2.micro for the CA nodes. It can be seen that the total cost is slightly under 50USD$. A cost analysis comparison of public Cloud computing versus on-premises hardware resources can be found in the work by Chandra et al. [31].

7. Conclusions

This paper has presented a Project Based Learning PBL driven by a Cloud-based Tool to collaboratively create a distributed Grid deployment, on top of virtualised resources, and to carry out a complete training covering learning objectives related to the development of Grid applications and the management and configuration of Grid resources and services. The PBL has some imbalances of complexity among the stages of the PBL but we can conclude that are reasonable and acceptable and feasible for being addressed in the next courses. This experience can be extrapolated to other courses that address distributed computing and that can benefit from an on-demand deployment of configurable computing resources on a Cloud site. The use of Grid as a Service (GaaS) had enabled to carry out educational activities in the context of a PBL from the point of view of both Grid users and Grid administrators, the latter was not possible in the previous approach using existing VOs and production platforms. This Grid as a Service (GaaS) tool has been built on top of ODISEA, an

[13] Amazon Web Services: `https://aws.amazon.com/`

Table 11: Cost per EA of each EA planned in the PBL. The cost has been computed for 36 students. (R) Running, (P) Paused.

EA	Time (Hours)	Week	VOMS Node	CA Node	Globus Node	Ad. Globus Node	AWS cost ($)
EA1	0.50	1	1 (R)	0	0	0	0.05
EA2	1.50	1	1 (P)	36 (R)	0	0	1.08
EA3	0.50	1	1 (R)	36 (P)	0	0	0.05
EA4	0.75	1	1 (R)	36 (R)	36 (R)	0	2.57
EA5	0.75	1	1 (P)	36 (R)	36 (R)	0	2.52
EA6	0.50	1	1 (P)	0	36 (R)	0	1.80
EA7	1	2	1 (R)	0	36 (R)	0	1.85
EA8	1	2	1 (P)	0	36 (R)	0	1.80
EA9	2.5	2	1 (R)	0	36 (R)	0	5.18
EA10	2	3	1 (P)	0	36 (R)	0	3.60
EA11	2.5	3	1 (R)	0	36 (R)	0	5.18
EA12	2	4	1 (P)	0	36 (R)	0	3.60
EA13	2.5	4	1 (P)	0	36 (R)	0	5.04
EA14	2	5	1 (R)	0	36 (R)	20 (R)	5.70
EA15	1.5	5	1 (R)	0	36 (R)	20 (R)	5.70
EA16	1	5	1 (R)	0	36 (R)	20 (R)	2.85
						TOTAL	48.57

open-source platform based on the Infrastructure Manager, that allows deploying and scaling customised virtual infrastructures. The economic cost assessed using Amazon Web Services indicates that performing the educational activities in a public Cloud provider is both cost-effective and convenient since no investment in hardware is required and no maintenance of the hardware infrastructure is needed by the educational institution. It is clear that combining learning methodologies based on PBLs conducted on a Cloud platform that provisions the computing and storage resources leads to a winning combination in terms of student satisfaction and final assessment results. As future work we plan to evolve towards immutable infrastructures using pre-configured Docker images for the different node types. This would significantly speed up the deployment process of the infrastructure, especially if using an on-premises Docker registry to achieve faster downloads of the Docker images in the different Virtual Machines.

8. Acknowledgement

The authors wish to thank the financial support received from Vicerrectorado de Estudios, Calidad y Acreditación of the Universitat Politècnica de València to develop the PIME project with reference B29, and to the Conselleria d'Educació, Investigació, Cultura i Esport de la Generalitat Valenciana for the project with reference number AICO/2019/303.

References

[1] M. L. Bote-Lorenzo, Y. A. Dimitriadis, E. Gómez-Sánchez, Grid characteristics and uses: a grid definition, in: European Across Grids Conference, Springer, 2003, pp. 291–298.

[2] I. Foster, C. Kesselman, The Grid 2: Blueprint for a new computing infrastructure, Elsevier, 2003.

[3] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke, The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets, Journal of network and computer applications 23 (3) (2000) 187–200.

[4] M. Chillarón, V. Vidal, D. Segrelles, I. Blanquer, G. Verdú, Combining grid computing and docker containers for the study and parametrization of ct image reconstruction methods, Procedia Computer Science 108 (2017) 1195–1204.

[5] A. Forti, S. Bavikadi, C. Bigongiari, G. Cabras, A. De Angelis, B. De Lotto, M. Frailis, M. Hardt, H. Kornmayer, M. Kunze, et al., Grid services for the magic experiment, in: Frontiers of Fundamental Physics, Springer, 2006, pp. 333–337.

[6] S. Bozhko, G. Asher, R. Li, J. Clare, L. Yao, Large offshore dfig-based wind farm with line-commutated hvdc connection to the main grid: Engineering studies, IEEE transactions on energy conversion 23 (1) (2008) 119–127.

[7] J. Austin, R. Davis, M. Fletcher, T. Jackson, M. Jessop, B. Liang, A. Pasley, Dame: Searching large data sets within a grid-enabled engineering application, Proceedings of the IEEE 93 (3) (2005) 496–509.

[8] I. Foster, C. Kesselman, S. Tuecke, The anatomy of the grid: Enabling scalable virtual organizations, The International Journal of High Performance Computing Applications 15 (3) (2001) 200–222.

[9] A. L. Bazinet, D. S. Myers, J. Fuetsch, M. P. Cummings, Grid services base library: A high-level, procedural application programming interface for writing globus-based grid services, Future Generation Computer Systems 23 (3) (2007) 517–522.

[10] M. David, G. Borges, J. Gomes, J. Pina, I. C. Plasencia, E. Fernández-del Castillo, I. Díaz, C. Fernandez, E. Freire, Á. Simón, et al., Validation

of grid middleware for the european grid infrastructure, Journal of grid computing 12 (3) (2014) 543–558.

[11] E. Laure, A. Edlund, F. Pacini, P. Buncic, M. Barroso, A. Di Meglio, F. Prelz, A. Frohner, O. Mulmo, A. Krenek, et al., Programming the grid with glite, Tech. rep. (2006).

[12] J. Walsh, J. Dukes, Application support for virtual gpgpus in grid infrastructures, in: 2015 IEEE 11th International Conference on e-Science, IEEE, 2015, pp. 67–77.

[13] S. Bell, Project-based learning for the 21st century: Skills for the future, The clearing house 83 (2) (2010) 39–43.

[14] D. Petcu, Teaching grid technologies to phd students, part 1: Using best practices to build the course, IEEE Distributed Systems Online 9 (3) (2008) 2–2.

[15] G. Andronico, V. Ardizzone, R. Barbera, R. Catania, A. Carrieri, A. Falzone, E. Giorgio, G. L. Rocca, S. Monforte, M. Pappalardo, et al., Gilda: The grid infn virtual laboratory for dissemination activities, in: First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities, IEEE, 2005, pp. 304–305.

[16] M. Fargetta, D. Scardaci, R. Barbera, Gilda status and recent activities in grid training, in: Managed Grids and Cloud Systems in the Asia-Pacific Research Community, Springer, 2010, pp. 311–322.

[17] R. Berlich, M. Hardt, M. Kunze, M. Atkinson, D. Fergusson, Egee: building a pan-european grid training organisation, in: ACM International Conference Proceeding Series, Vol. 167, Citeseer, 2006, pp. 105–111.

[18] D. Gorgan, T. Stefanut, V. Bacu, Grid based training environment for earth observation, in: International Conference on Grid and Pervasive Computing, Springer, 2009, pp. 98–109.

[19] C. Cherubino, L. Ciuffo, A. Fuentes, R. Mayo, et al., Eela training activities in spain, in: Proceedings of the Spanish Conference on e-Science Grid Computing, March 1-2, 2007. Madrid (Spain), 2007.

[20] I. Gregori, M. Patil, S. Cozzini, Gridseed: A virtual training grid infrastructure, Joint Eu-IndiaGrid/CompChem Grid Turorial on Chemical an dMaterial Science Applications (2011) 39–53.

[21] S. Childs, B. Coghlan, J. McCandless, Gridbuilder: A tool for creating virtual grid testbeds, in: 2006 Second IEEE International Conference on e-Science and Grid Computing (e-Science'06), IEEE, 2006, pp. 77–77.

[22] R. M. Capraro, M. M. Capraro, J. R. Morgan, Stem project-based learning, An Integrated Science, Technology, Engineering, and Mathematics (STEM) Approach 2.

[23] M. Feller, I. Foster, S. Martin, Gt4 gram: A functionality and performance study, in: TeraGrid Conference, Citeseer, 2007.

[24] J. Bester, I. Foster, C. Kesselman, J. Tedesco, S. Tuecke, Gass: A data movement and access service for wide area computing systems, in: Proceedings of the Sixth workshop on I/O in Parallel and Distributed Systems, 1999, pp. 78–88.

[25] J. Bresnahan, M. Link, G. Khanna, Z. Imani, R. Kettimuthu, I. Foster, Globus gridftp: what's new in 2007, in: Proceedings of the first international conference on networks for grid applications, 2007, pp. 1–5.

[26] X. Zhang, J. M. Schopf, Performance analysis of the globus toolkit monitoring and discovery service, mds2, in: IEEE International Conference on Performance, Computing, and Communications, 2004, IEEE, 2004, pp. 843–849.

[27] J. D. Segrelles, G. Moltó, M. Caballer, Remote computational labs for educational activities via a cloud computing platform, in: 2015 Proceedings

of the Information Systems Education Conference (ISECON), 2015, pp. 309–321.

[28] J. D. Segrelles, G. Moltó, Assessment of cloud-based computational environments for higher education, in: 2016 IEEE Frontiers in Education Conference (FIE), IEEE, 2016, pp. 1–9.

[29] J. A. González-Martínez, M. L. Bote-Lorenzo, E. Gómez-Sánchez, R. Cano-Parra, Cloud computing and education: A state-of-the-art survey, Computers & Education 80 (2015) 132–151.

[30] M. R. Sheldon, M. J. Fillyaw, W. D. Thompson, The use and interpretation of the friedman test in the analysis of ordinal-scale data in repeated measures designs, Physiotherapy Research International 1 (4) (1996) 221–228.

[31] D. G. Chandra, M. D. Borah, Cost benefit analysis of cloud computing in education, in: 2012 International Conference on Computing, Communication and Applications, IEEE, 2012, pp. 1–6. doi:10.1109/ICCCA.2012.6179142.