



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



DEPARTAMENTO DE SISTEMAS
INFORMÁTICOS Y COMPUTACIÓN

Departamento de sistemas Informáticos y
Computación
Universitat Politècnica de València

Despliegue de infraestructuras Grid en la nube

Trabajo Fin de Máster

**Máster Universitario en Computación Paralela y
Distribuida**

Autor: Jesús Juan González Nieto

Tutor: J. Damián Segrelles Quilis

Curso Académico 2016-2017

Resumen

Un Grid se compone de un conjunto de recursos de cómputo y datos ubicados en dominios administrativos diferentes que se gestionan bajo el marco de una Organización Virtual (OV) con el objeto de resolver problemas científicos.

Existen OV específicas tanto a nivel nacional (e.g. tut.vo.ibergrid.eu) como a nivel internacional (Testbed Gilda) que proporcionan infraestructuras Grid específicas para uso docente. Sin embargo, estas OVs tienen un uso limitado para la realización de actividades educativas, debido a que éstas solo se pueden utilizar desde un punto de vista de un usuario Grid y no como administradores de recursos, dado que son recursos prefijados, imposibilitando agregar o borrar nuevos de forma elástica y dinámica.

En este trabajo se presenta un recurso docente que despliega dinámica y elásticamente un Grid as a Service (GaaS) en la nube, utilizando tanto proveedores públicos (Amazon Web Services) como privados (Open-Nebula), en la que se virtualizan los dominios administrativos del Grid y se integran en una OV. Estos GaaS se crean con fines educativos y se pueden emplear tanto para formar usuarios del Grid como administradores.

Palabras clave: Grid, Cloud, Virtualización, Aprendizaje Basado en Proyectos

Abstract

A Grid is composed by a set of computational and data resources located in different administrative domain that are managed under the framework of a Virtual Organization (VO) and aim to resolve scientific problems.

There are specific VOs both national (VO tut.vo.ibergrid.eu) and international (Gilda testbed) that provide specific Grid infrastructures for educational purposes. However, these VOs have limited interest for educational activities since they can only be used from the Grid user point of view and not as administrators of the resources and services. This is because they use preallocated resources and it is impossible add or remove resources in the existing infrastructure in a dynamic and elastic way.

In this work we present an educational resource that dynamically deploys a Grid as a Service (GaaS) on the Cloud, using both public providers (Amazon Web Services) and on-premises (OpenNebula), virtualizing the administrative domains of the Grid that are integrated in a VO. These GaaS are used for educational purposes and can be employed for teaching both Grid users and system administrators.

Keywords: Grid, Cloud, Virtualization, Project-based learning.

Agradecimientos

Quisiera hacer una pequeña mención a diversas figuras que han sido de gran apoyo en todo el transcurso de la realización de este trabajo final de máster. En primer lugar, mi familia, porque siempre ha estado apoyándome y haciéndome más fácil las horas encerrado frente al ordenador.

Después dar las gracias a Amparo por todo el apoyo, la paciencia y el cariño con el que me ha dado fuerzas para seguir adelante cuando he tenido dificultades. No podría haber aguantado muchas largas noches sin sus ánimos.

Tras esto agradecer a todos los amigos que con pequeños gestos de ayuda tales como resolver un problema o dar un consejo me han ayudado tanto a lo largo de la realización. Y a todos los que aun no sabiendo cómo ayudarme me mandaban sus ánimos porque sabían que los necesitaba.

También agradecer a Damián, mi tutor, por toda la paciencia que ha tenido conmigo. Porque ha sido mi guía y gracias a él, he podido aprender mucho más de lo que esperaba.

Por último y no menos importante, agradecer al i3m por brindarme un puesto en el que trabajar dentro de la universidad para poder escapar de la comodidad de casa que no siempre es muy productiva.

Índice de contenidos

1. Introducción	10
2. Objetivos	12
3. Estructura de la obra	14
4. Estado del arte	15
4.1. Infraestructuras grid para docencia	15
4.1.1. NGI (Tut.vo.ibergrid.eu)	15
4.1.2. GILDA.....	16
4.2. Middleware Grid	17
4.2.1. UNICORE.....	17
4.2.2. ARC	17
4.2.3. GLite	18
4.2.4. Globus Toolkit.....	18
4.3. Cloud	20
4.3.1. Infrastructure manager (IM)	20
4.3.2. Proveedores Cloud On-premise	22
4.3.3. Proveedores Cloud Públicos.....	24
5. Recurso Docente	27
5.1. Análisis de requisitos.....	27
5.1.1. Escenario de referencia.....	27
5.1.2. Requisitos funcionales.....	29
5.1.3. Requisitos no funcionales.....	30
5.2. Arquitectura	31
5.3. Flujo de trabajo	37



Grid As A Service: Herramienta para el despliegue y gestión de un Grid en la nube para actividades educativas

5.3.1.	Inicialización	37
5.3.2.	Adición de nodos.....	38
5.3.3.	Borrado de nodos.....	40
5.3.4.	Finalización.....	41
6.	Pruebas y resultados	42
7.	Trabajos futuros	45
8.	Conclusiones.....	46
9.	Bibliografía	47



Índice de ilustraciones

Ilustración 1 - Arquitectura de Globus Toolkit	19
Ilustración 2 - Ejemplo de receta RADL	21
Ilustración 3 - Arquitectura del IM	22
Ilustración 4 - Servicios ofrecidos por Open Nebula	23
Ilustración 5 - Arquitectura de OpenStack	24
Ilustración 6 - Mapa de la arquitectura de servicios de AWS.....	25
Ilustración 7 - Arquitectura de Microsoft Windows Azure	26
Ilustración 8 - Mapa de servicios de Google Compute Engine	27
Ilustración 9 – Arquitectura inicial de la aplicación.....	32
Ilustración 10 – Arquitectura de comunicaciones.....	33
Ilustración 11 - Arquitectura de un nodo FE	34
Ilustración 12 - Arquitectura de un nodo CA.....	35
Ilustración 13 - Arquitectura de un nodo Globus.....	36
Ilustración 14 - Diagrama del flujo de la inicialización	37
Ilustración 15 - Diagrama del flujo de la adición de nodos	39
Ilustración 16 - Diagrama de flujo de la eliminación de nodos	40
Ilustración 17 - Gráfica de tiempos de despliegue por nodos.....	43
Ilustración 18 - Gráfico de tiempo total del despliegue de una infraestructura.....	44

Índice de tablas

Tabla 1 - Recursos OV Tut.vo.ibergrid.eu	16
Tabla 2 - Recursos OV GILDA	16
Tabla 3 - Unidades temáticas	29

1. Introducción

Un Grid [8], en el área de la computación distribuida, se define como un conjunto de recursos heterogéneos (cómputo, redes, software y datos) y servicios compartidos entre diferentes dominios administrativos (universidades, institutos de investigación, etc...). Todos estos recursos y servicios, se integran y gestionan bajo el concepto de Organización Virtual [8] (OV), la cual proporciona a los investigadores un entorno desde donde pueden acceder a un gran número de recursos [5]. Las OV permiten llevar a cabo a una mayor escala experimentos que requieren de una elevada carga computacional y de almacenamiento de datos a través de sus servicios, alcanzando tiempos de computación más reducidos respecto a los que se obtendrían utilizando recursos locales. Existen ejemplos del uso del Grid para la resolución de problemas científicos en diferentes áreas de la ciencia, tales como la radiología [1], astronomía [7] o la física de altas energías [5], entre muchas otras.

Los servicios que proporciona un Grid a través de las OV a la comunidad científica, se ofrecen mediante middlewares [10] (APIs o Command Lines Interfaces) o gateways [13] construidos sobre estos. Estos middlewares permiten crear y gestionar Infraestructuras avanzadas Grid (IAGs), abstrayendo al usuario final de la complejidad de manejo y orquestación de los diferentes recursos ofrecidos por el Grid (Ej. seleccionar los recursos más apropiados del Grid para la ejecución de una tarea/trabajo en base a unos requisitos). Principalmente, los servicios que ofrecen los middlewares se encargan de gestionar:

- a) aspectos de seguridad relativas a la autenticación y autorización de los usuarios Grid a los recursos y servicios de una OV.
- b) el ciclo de vida completo de los trabajos a ejecutar en el Grid.
- c) los datos a través de catálogos.
- d) la monitorización y descubrimiento de recursos.

Alguno de los middlewares Grid más conocidos para gestionar infraestructuras avanzadas son UNICORE [6], Advanced Resource Connector (ARC) [14] o gLite [12]. Con el objeto de consolidar, armonizar y dar soporte a estos tres middleware y converger a un único

middleware, se creó la plataforma European Middlewre Initiative (EMI¹). Otros middleware como Globus [3], son de más bajo nivel de abstracción y tienen como objetivo proporcionar herramientas básicas para construir middleware como los integrados en EMI. De hecho, parte de gLite ha sido desarrollado utilizando la toolkit de globus.

En la actualidad, existen IAGs desplegadas utilizando los middlewares comentados. A nivel europeo, las más conocidas son las National Grid Infrastructures (NGIs), las cuales se despliegan bajo los servicios ofrecidos por el middleware gLite, y en la que cada NGI se organiza dentro del país o, en el caso de España, abarcando la península ibérica (España y Portugal). Las NGIs comparten sus recursos a través de diferentes OV's creadas para áreas científicas específicas y se coordinan a través de la European Grid Infrastructure [11] (EGI) que de forma transversal integra todos los recursos existentes a través de un conjunto de OV europeas, como por ejemplo en el área de biomedicina, a través de la OV *biomed*.

La gran mayoría de las OV proporcionadas por las NGI y EGI son creadas para fines científicos, dado que son para lo que fueron creadas estas IAGs. Sin embargo, también existen OV como *tut.vo.ibergrid.eu* creadas para la realización de actividades educativas relacionadas con los servicios que puede ofrecer y cómo estas pueden ser utilizadas para la realización de estudios científicos mediante el desarrollo de aplicaciones Grid. En este sentido, existe también el Grid INFN Virtual Laboratory for Dissemination Activities [2] (GILDA) el cual es una infraestructura Grid que también utiliza gLite y fue creada específicamente para demostrar y diseminar las capacidades y posibilidades que ofrece el Grid computing para la ciencia, principalmente para investigadores del área de las físicas de altas energías. La infraestructura de GILDA está desplegada principalmente por Italia, aunque cuenta también con recursos repartidos en diferentes puntos de todo el mundo.

Este TFM se centra en la parte educativa de las IAGs, con el objeto de cubrir, desde el punto de vista educativo, algunas de las necesidades de las OV docentes existentes que no están cubiertas. Las IAGs, tales como las NGIs o GILDA, a través de sus OV de docencia permiten formar a los investigadores para la creación de aplicaciones científicas Grid. Sin embargo, una de las carencias de estos entornos es que se utilizan solo para formar a científicos e investigadores en el uso de un Grid a nivel de usuario y no en la parte de la instalación, configuración y administración de los recursos Grid de una OV, siendo esta parte fundamental para que desarrolladores bien formados en el área de Grid Computing puedan crear nuevos servicios que hagan más eficientes y productivas las IAGs ya existentes. Por ello, en este TFM se plantea crear

¹ <http://www.eu-emi.eu>

un recurso docente a través de una plataforma que permita formar tanto a los científicos desde el punto de vista de usuarios del Grid, como desde el punto de vista de administradores de recursos. Por ello, es fundamental proporcionar mecanismos que permitan crear y desplegar desde cero infraestructuras que emulen IAGs, de tal manera que los administradores conozcan y dominen las tecnologías y aspectos técnicos subyacentes a la configuración y despliegue de recursos Grid. La creación de una IAG desde cero, implica disponer de acceso a múltiples nodos físicos que emulen dominios administrativos diferentes, y que, a su vez, estos estén conectados, por lo que se requiere de una configuración de redes específicas que combine las redes privadas de los diferentes dominios administrativos con una red pública (internet) que los conecte. Para ello, se requiere de un acceso a cada uno de los nodos en modo administrador, para ir configurándolos con el fin de emular los dominios mediante el uso de un middleware Grid y posteriormente conectarlos a través de una red pública. Preparar este tipo de entorno, exige mucho tiempo de configuración, y más si consideramos que el número de dominios administrativos puede variar de forma dinámica en un Grid

Una forma de amortiguar este coste de configuración podría ser mediante la creación de un IAG virtual utilizando las tecnologías Cloud. Parece que el Cloud Computing sea una buena alternativa para poder generar Grids virtualizados, en la que se simulan los diferentes dominios administrativos que lo conforman y los recursos que en ellos se administran. Cloud computing permite de una forma dinámica y elástica aprovisionar estas infraestructuras y configurarlas de forma automatizada, siendo estas ofrecidas como un servicio por los proveedores Cloud que los ofrecen. Dichos proveedores pueden ser tanto públicos (e.g. AWS², Azure³, Compute Engine⁴) como on-premise cuando se gestionan de forma privada a través de gestores como OpenNebula o OpenStack. En los últimos años, se han publicado diferentes trabajos que debaten sobre las ventajas e inconvenientes del Cloud computing [9] tanto desde el punto de vista de los centros educativos [15], como del profesorado [16] como de los estudiantes [15].

2. Objetivos

² <https://aws.amazon.com/>

³ <https://azure.microsoft.com/>

⁴ <https://cloud.google.com/compute/>

El objetivo de este TFM es el de diseñar e implementar un recurso docente a través de una plataforma donde se proporcione una IAG como un servicio en la nube (Grid as a Service - GaaS), de forma que el docente pueda desplegar y replugar dinámicamente infraestructuras Grid ad-hoc específicamente configuradas para la realización de Actividades Educativas (AEs) tanto desde el punto de vista de usuario Grid (ej. desarrollo de aplicaciones Grid que resuelvan problemas científicos), como desde el punto de vista de un administrador de recurso Grid (Ej. creación de una IAG desde cero o la integración de nuevos recursos Grid en una ya existente).

Para alcanzar este objetivo general, se definen los siguientes tres objetivos específicos:

- Estudio del escenario de referencia que permita extraer los requisitos (funcionales y no funcionales) del recurso docente a desarrollar. Dicho escenario de referencia debe de abordar aquellas AEs requeridas tanto desde el punto de vista del usuario Grid como del administrador de recursos.
- Diseño e implementación de la arquitectura del recurso docente de forma que cumpla con los requisitos extraídos del escenario de referencia.
- Automatizar los procesos de configuración de los nodos de la infraestructura para que el profesor tenga que realizar las mínimas interacciones y tener una infraestructura preparada para la realización de las AEs por parte de los alumnos tanto desde el punto de vista de un usuario Grid que desarrolla aplicaciones científicas sobre una IAG, como desde el punto de vista de un administrador de recursos Grid.
- Evaluar el recurso docente en términos de tiempos de despliegue.

3. Estructura de la obra

El trabajo se compone de los siguientes capítulos. En la sección 1 se introduce el trabajo comentando un pequeño análisis de la realidad en la que actúa. En la sección 2 se presentan los objetivos que quiere conseguir el recurso docente planteado. En la sección 3 en el que nos encontramos, se describe la estructura que sigue la obra. En la sección 4 se analiza el estado del arte en el ámbito de este TFM centrándose en los siguientes aspectos: infraestructuras grid para la docencia, middleware grid infraestructuras y proveedores cloud.

Tras este pasamos a la sección central del trabajo, el 5. En esta sección se describe toda la estructura del recurso docente desarrollado. Desde el análisis de requisitos hasta el seguimiento del flujo de trabajo. La siguiente, la sección 6, expone las pruebas realizadas y los resultados obtenidos. Y para finalizar tenemos la sección 7 que son las posibles mejoras del recurso desarrollado, y la sección 8 donde se exponen las conclusiones.

4. Estado del arte

A continuación se va a proceder a analizar 3 apartados del trabajo que han dado lugar a la elección de un tipo de herramienta para el desarrollo del recurso docente. Primero las infraestructuras grid para la docencia, comentando brevemente algunas de las más relevantes en la actualidad, y como el recurso desarrollado busca corregir las carencias de estas. Otro apartado es el middleware grid, realizando un pequeño repaso por las actuales opciones disponibles en el mercado, sus características y porque se ha elegido Globus toolkit. Y por último se hará un rápido análisis a los diferentes proveedores de infraestructuras cloud que existen, comentando sus características principales y su impacto a la hora de escoger una para el desarrollo de este trabajo.

4.1. Infraestructuras grid para docencia

EL objetivo de este TFM gira en torno al despliegue de IAGs para la docencia. Por ello, se ha realizado un pequeño estudio de dos IAGs que proporcionan OV específicas para la realización de AEs. La primera es la OV **tut.vo.ibergrid.eu**, perteneciente al NGI de la península ibérica, y la segunda el Grid Infn Laboratory for Dissemination Activities (GILDA) perteneciente al Italian Grid Infrastructure (IGI).

4.1.1. NGI (Tut.vo.ibergrid.eu)

Esta es una OV ofrecida por el NGI ibérica para la realización de tutoriales, cursos y, en general, para el ámbito de la educación alrededor de la computación en grid. El NGI cuenta también con otras OV no docentes, pensadas para el desarrollo de aplicaciones grid en varios campos de la ciencia, por ejemplo, la OV **phys.vo.ibergrid.eu** para física y ciencia aeroespacial, **eng.vo.ibergrid.eu** para ingenierías, entre otras. La **tut.vo.ibergrid.eu**, es una OV que se pone a la disposición (bajo petición de acceso) a cualquier usuario, y ponen a disposición de sus usuarios una IAG para que se desarrollen AEs desde el punto de vista de un usuario Grid que necesite desarrollar una aplicación Grid, con el fin de aprender cómo funcionan estas IAGs y los servicios que ofrece.

Los recursos que ofrece esta VO se resumen en la siguiente tabla:

Nombre OV:	tut.vo.ibergrid.eu
Nodos de cómputo (CE)	10
Nodos de almacenamiento (SE)	3
Capacidad total de almacenamiento	585176
Nodos VOMS	4
Nodos catálogo (LFC)	4
Nodos WMS	4

Tabla 1 - Recursos OV Tut.vo.ibergrid.eu

4.1.2. GILDA

El instituto nacional de física nuclear italiano desarrolló GILDA como herramienta para la adquisición de conocimientos relativos al uso de IAGs, puesto que las tecnologías grid avanzaban muy rápido. Esta infraestructura permite realizar test y probar todas las capacidades que ofrece la computación en grid.

Al igual que la OV tut.vo.ibergrid.eu tiene el inconveniente de que ofrece múltiples herramientas y entornos, pero todos desde el punto de vista del usuario Grid para el desarrollo de aplicaciones científicas. En ninguna de las IAGs estudiadas se permite interactuar con el grid desde un punto de vista de administrador.

Nombre OV:	GILDA
Nodos de cómputo (CE)	5
Nodos de almacenamiento (SE)	3
Capacidad total de almacenamiento	88979
Nodos VOMS	2
Nodos catálogo (LFC)	1
Nodos WMS	2

Tabla 2 - Recursos OV GILDA

El recurso docente a desarrollar en este TFM, busca una gestión propia de la IAG, donde se puedan trabajar además de los aspectos relativos a la creación de aplicaciones Grid científicas, los aspectos de administración de recursos Grid tales como como la gestión de nodos que actuarán como recursos de almacén o cómputo en el Grid, configuración del middleware e incluso la interacción con las Entidades de Autorización (CAs) para las conseguir credenciales requeridas para una adecuada configuración de los recursos. Todo esto ayudará a conseguir una

experiencia docente más completa respecto a las que se pueden conseguir en las IAGs comentadas en este apartado.

4.2. Middleware Grid

Para desarrollar una plataforma que permita gestionar el despliegue de una nueva IAG, primero es necesario elegir qué middleware grid se va a utilizar como base para el funcionamiento de la IAG. En el mercado actual, existen varias alternativas tales como ARC, UNICORE, gLite o Globus Toolkit. En este TFM nos centraremos en estos cuatro, puesto que son los más populares y utilizados. A continuación, se describen resumidamente las características de ambos middlewares.

4.2.1. UNICORE

UNICORE⁵ (UNiform Interface to COmputing REsources) es un middleware grid que fue desarrollado por el Ministerio Federal de Educación e Investigación de Alemania como solución a la demanda de una herramienta con la que poder explotar todos los recursos de cómputo de las diferentes ciudades alemanas con *data centers* a través de internet.

Desde esta idea se fue desarrollando para convertirse en una alternativa al Globus Toolkit. Su arquitectura se compone de 3 capas [17].

- Una capa cliente que gestiona la creación y gestión de los trabajos lanzados.
- Una capa servidor que es un control de seguridad para el flujo de trabajos mediante el uso de certificados.
- Una capa llamada *Target System Interface* que reside en el host que ejecuta los trabajos. Es un pequeño demonio que gestiona los trabajos a nivel local.

Este middleware no es el escogido principalmente porque es un producto que lleva todas las herramientas para hacer funcionar un grid, no como Globus que está pensado como una toolkit para desarrollar middlewares. UNICORE es un middleware que puede trabajar utilizando Globus como base, pero que no está pensado para desarrollar sobre este. Y puesto que en este trabajo se busca un middleware sobre el que desplegar una infraestructura con fines educativos, este middleware no se ha utilizado.

4.2.2. ARC

ARC (*Advanced Resource Connector*) es un middleware de computación grid desarrollado por Norbugrid en 2004. Esta organización está formada por un grupo de

⁵ <https://www.unicore.eu/>



investigadores de diferentes lugares que mantienen un acuerdo de colaboración para desarrollar el middleware grid ARC. Este nace de la necesidad de una solución software para la computación a gran escala por parte de la comunidad científica nórdica. Para ello se apoyaron en el estándar de facto en este campo, Globus toolkit, en su versión 2 (en el momento del desarrollo inicial), utilizando herramientas de este como el GSI (Globus Security Infrastructure) y el protocolo GridFTP para la gestión de ficheros. ARC proporciona una serie de servicios y herramientas para poder gestionar y explotar una gran cantidad de recursos con fines principalmente de investigación. Es comúnmente utilizada en el campo de la física, y fue el primer middleware en proporcionar servicios de cómputo grid en el campo de la física de altas energías.

Este middleware ha sido descartado para el uso en el recurso docente debido a que es un middleware cuyo objetivo es el campo de la investigación y trabaja a un alto nivel. Y en este trabajo se busca un software de bajo nivel sobre el que poder aprender todos los aspectos del grid.

4.2.3. GLite

GLite es un middleware que soporta la creación de infraestructuras grid a gran escala con un alto número de recursos y capacidad de cómputo. Nacido del proyecto *Enabling Grids for E-sciencE* (EGEE⁶) fue hospedado por el CERN y actualmente forma parte del *Unified Middleware Distribution* (UMD⁷) de la *European Grid Infrastructure* (EGI⁸), la cual reúne diversos componentes de software para la computación en el grid. Un problema que tiene este middleware es la escasa compatibilidad con los sistemas operativos, ya que solo es compatible con CentOS y Scientific Linux.

Este middleware ofrece una solución de muy alto nivel para lo que se busca en este trabajo. Ya que se quiere desarrollar, en el ámbito educativo, tanto el perfil de usuario, como el de administrador del grid, este middleware es descartado en el recurso docente planteado.

4.2.4. Globus Toolkit

Globus toolkit es un middleware que nació en 1998 de un proyecto liderado por Ian Foster y llevado a cabo entre el Argonne National Laboratory y la Universidad de Chicago. Desde entonces ha avanzado mucho (la versión estable en el momento de desarrollo de esta

⁶ <http://cern.ch/egee/>

⁷ <http://repository.egi.eu/>

⁸ <https://www.egi.eu/>

plataforma es la 6) y en la actualidad es un software referencia en el campo de la computación en grid.

Este middleware se compone de un conjunto de bibliotecas, servicios y APIs para dotar de una amplia funcionalidad al grid. Un punto a favor de este paquete es que, al contrario que otros como gLite, tiene una amplia compatibilidad. Pudiéndose instalar tanto en Linux, como en Windows o Mac OS X. Además, tiene versiones de 32 y 64 bits. Glite, por ejemplo, solo disponía de una distribución de 64 bits.

A continuación, podemos observar en la ilustración 1, una imagen descriptiva con los diferentes módulos que conforman globus, y aunque es una referencia de la anterior versión (Globus Toolkit 5.0) no se ha cambiado la estructura de módulos en la versión 6.

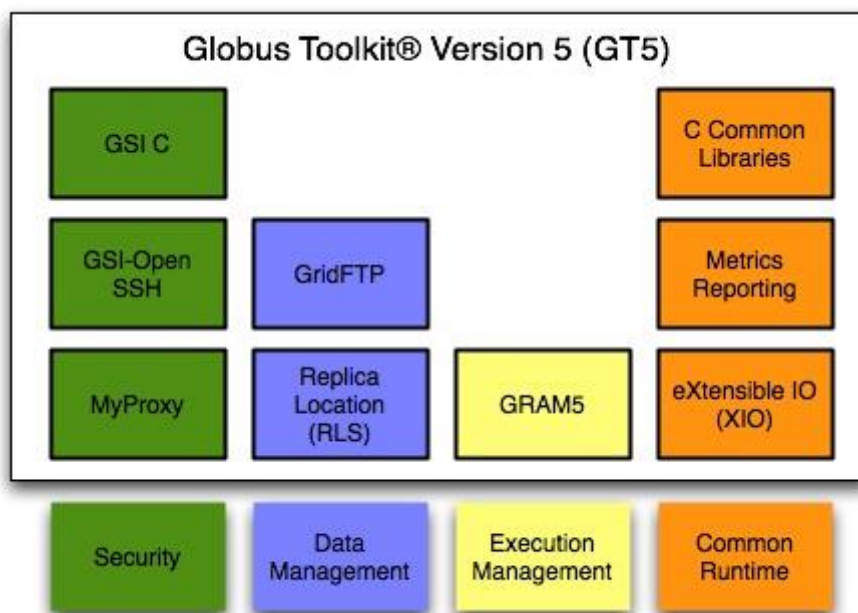


Ilustración 1 - Arquitectura de Globus Toolkit

Como podemos observar en la imagen, están presentes los 3 puntos que se han descrito en la introducción cuando se habla de un middleware en computación grid. Por un lado la seguridad, con el control de acceso a la infraestructura mediante autorización por certificado y generación de proxies. Por otro lado, tenemos la gestión de datos que incluye la gestión de un catálogo de réplicas. Y el tercer punto descrito es el sistema de gestión de trabajos, del que se encarga el componente GRAM. A parte de estos módulos tenemos una sección con módulos auxiliares para mejorar el funcionamiento del grid, como herramientas para establecer métricas y para poder hacer un seguimiento del funcionamiento y monitorización del grid.

Debido a que es el middleware de más bajo nivel y con más recorrido en el campo de la computación grid, siendo un estándar de facto, se ha desarrollado el recurso docente que desarrolla este TFM con Globus toolkit.

4.3. Cloud

La base de este TFM es el uso de las tecnologías Cloud con el fin de crear IAGs como servicios en la nube. El cloud abre nuevos caminos debido a las ventajas que proporciona la virtualización. Por una parte, se pierden las restricciones de tener que usar máquinas físicas, así como el mantenimiento, coste o la dificultad a la hora de crear o eliminar una máquina de una infraestructura, dado que se delega al proveedor cloud.

En esta sección, primero se presenta el *Infrastructure Manager (IM)*, componente principal del recurso docente planteado en este TFM, que es una herramienta que permite el despliegue de máquinas virtuales y su contextualización en proveedores tanto públicos como privados (*On-Premise*). Además, en esta sección se da un breve repaso de los diferentes proveedores que se disponen para evaluar el recurso docente planteado en este TFM.

Existen proveedores cloud públicos y privados (*On-Premise*). A continuación, se analizarán las diferentes opciones que dan lugar al uso de uno u otro tipo de proveedor.

4.3.1. Infrastructure manager (IM)

El Infrastructure Manager [4] (IM⁹) es una herramienta desarrollada por el grupo de Grid y Computación de Altas Prestaciones de la Universidad Politécnica de Valencia (UPV). Dicha herramienta ha sido desarrollada con el fin de facilitar el acceso y el uso de diferentes proveedores de IaaS de una forma agnóstica, automatizando muchas de las operaciones que se realizan en estos para la gestión de máquinas virtuales (selección de imágenes, despliegue, configuración, instalación de software y actualización). Además, cuenta con un sistema de contextualización, basado en la herramienta de *DevOps* Ansible, que ayuda a instalar y configurar el software necesario en las máquinas desplegadas.

Esta herramienta funciona mediante unas recetas elaboradas en RADL (Resource and Application Description Language), un lenguaje para la especificación de requisitos a la hora de desplegar máquinas virtuales en el cloud. Estas recetas se componen de 4 apartados principales:

⁹ <http://www.grycap.upv.es/im>

- **Hardware.** Donde se detalla la especificación del hardware deseado para la MV a desplegar (CPU, Memoria RAM, etc.)
- **Software.** En este apartado se detallan las aplicaciones, librerías y otras herramientas software que se requieren en la máquina
- **Redes.** Mediante esta sección se puede especificar la arquitectura de redes que se desea. Utilizar una interfaz pública, una red privada, elegir nombre dns, etc.
- **Contextualización.** Esta es la sección que detalla una serie de instrucciones que configurarán la máquina para dejarla como el usuario requiera.

A continuación, en la ilustración 2, se puede observar un pequeño ejemplo de receta de un nodo Ubuntu en RADL:

```
network net (outbound = "yes")

system node_ubuntu (
  cpu.arch = 'i686' and
  memory.size >= 512M and
  net_interface.0.connection = "net" and
  net_interface.0.dns_name = "node-#N#" and
  disk.0.os.name = "linux" and
  disk.0.os.flavour = "ubuntu" and
  disk.0.os.version >= "12.04"
)

deploy node_ubuntu 1
```

Ilustración 2 - Ejemplo de receta RADL

En cuanto al soporte de proveedores, es otro punto por el que se ha escogido la utilización de esta herramienta para el desarrollo del recurso docente. El IM ofrece un soporte para el API de muchos proveedores cloud tanto públicos (Ej. AWS, Azure) como privados (Ej. OpenNebula, OpenStack), de modo que ayudara al recurso docente a dar más opciones al usuario en cuanto a la configuración del proveedor IaaS. A continuación se puede observar una imagen a modo de esquema de la arquitectura de la herramienta (Véase ilustración 3).

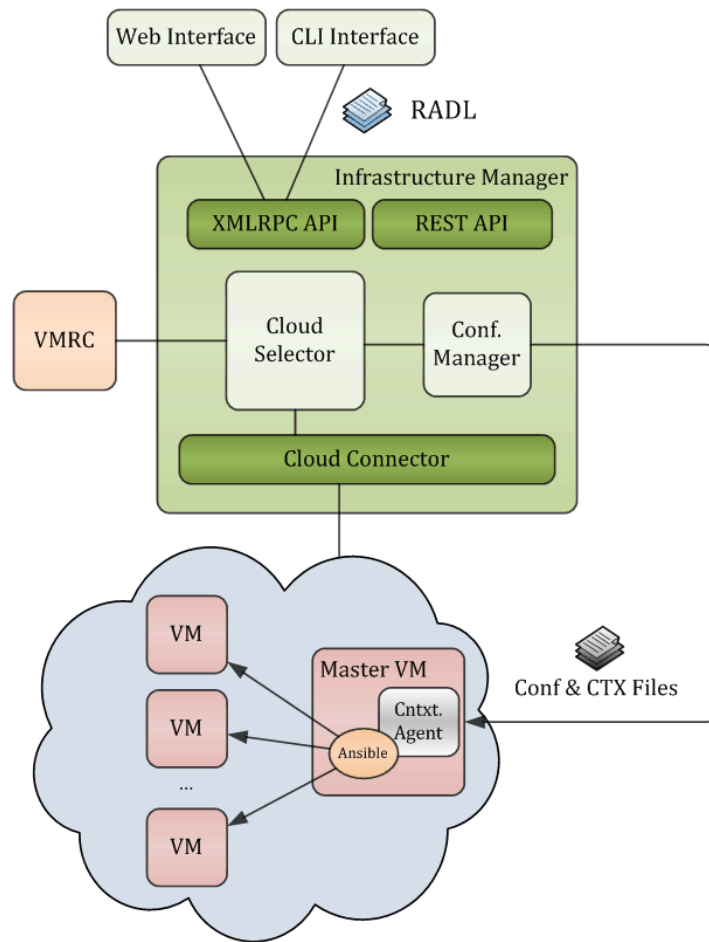


Ilustración 3 - Arquitectura del IM

4.3.2. Proveedores Cloud On-premise

Este tipo de cloud se utiliza cuando se posee y se quiere explotar una infraestructura física de forma local. Es un cloud enfocado a la optimización de recursos físicos locales mediante la gestión de despliegue y administración de máquinas virtuales.

Los *Cloud Management Platform* (CMP) son herramientas que ofrecen una capa de abstracción sobre la infraestructura física para gestionar todo el ciclo de vida de las máquinas virtuales que se despliegan sobre estos recursos.

Este tipo de cloud tiene unas ventajas con respecto a los de tipo público, por ejemplo, la personalización y la modificación de las máquinas es más fácil y con mayor amplitud en este modelo, ya que al disponer de acceso total a las máquinas hay más opciones que configurar. También la seguridad, ya que, al estar en una infraestructura local, el protocolo de seguridad se puede hacer desde cero y enfocarlo como cada organización desee. Esto es debido a que se tiene un mayor control, la infraestructura sobre la que se trabaja es propia y las políticas de seguridad no son impuestas por terceros. Por otro lado, tiene desventajas como la dificultad a

la hora de aumentar el número de máquinas o escalarlas, ya que en comparación con un cloud público, requeriría de mucho tiempo y no siempre sería posible. Esto es debido a que en una infraestructura local se cuenta con un número limitado de nodos, al contrario que en un proveedor cloud, que los recursos son “ilimitados” (no literalmente, pero no se limitan de la misma manera que en una red de recursos local). Estos cloud son los elegidos cuando se busca tener un control total sobre el despliegue de una aplicación, ya que todos los aspectos pueden ser parametrizados y no depende de un proveedor tercero.

Existen diversos CMP para la gestión y manejo clouds on-premise. A continuación se van a comentar dos de los más extendidos CMPs:

- **Open Nebula.** Desarrollado por la Universidad Complutense de Madrid (UCM) y OpenNebula Systems (conocido anteriormente como C12G Labs). Cuenta con el apoyo de muchas empresas y se utiliza comúnmente en la comunidad científica. Es un proyecto de código abierto y es fácilmente modificable para adaptarlo a las necesidades de cada realidad. A continuación en la ilustración 4 se puede ver un esquema de las principales funciones que ofrece OpenNebula.

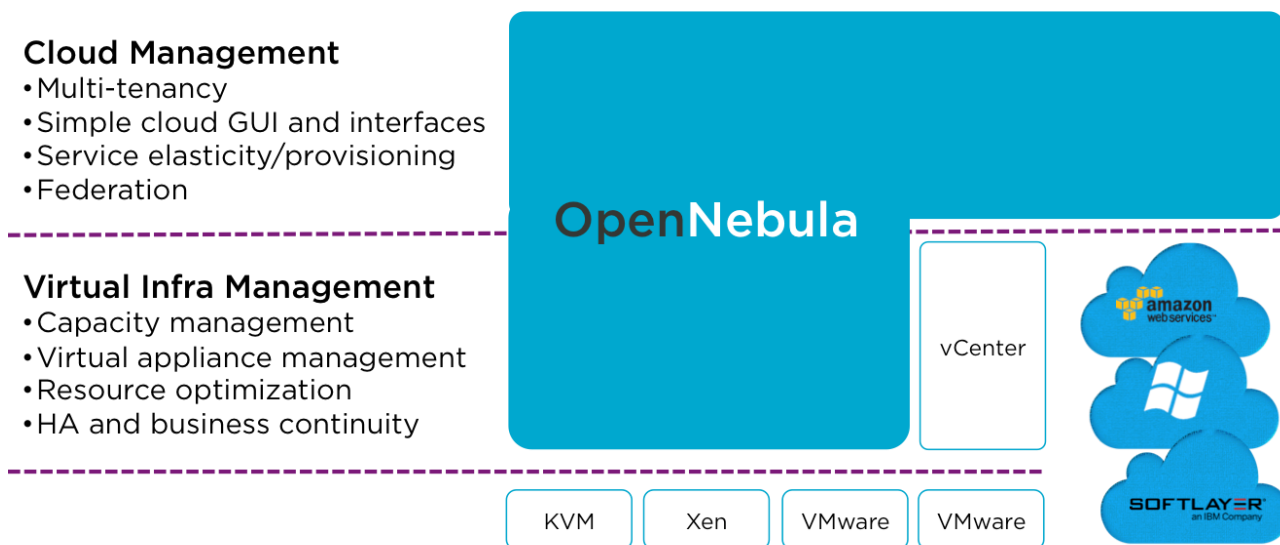


Ilustración 4 - Servicios ofrecidos por Open Nebula

- **Open Stack.** Es un proyecto de código abierto gestionado por la fundación OpenStack, al que se han unido empresas como AMD, Canonical, Cisco o Dell entre otras. Un proyecto que se desarrolla junto con la comunidad que tiene alrededor. Este proyecto busca una solución para gestionar grandes conjuntos de recursos de cómputo, de almacenamiento y de red a través de un *data center*. Y todo esto

Grid As A Service: Herramienta para el despliegue y gestión de un Grid en la nube para actividades educativas

gestionado desde una interfaz web. La estructura de este software es basada en módulos. En la ilustración 5 se puede ver un pequeño esquema de la arquitectura de OpenStack.

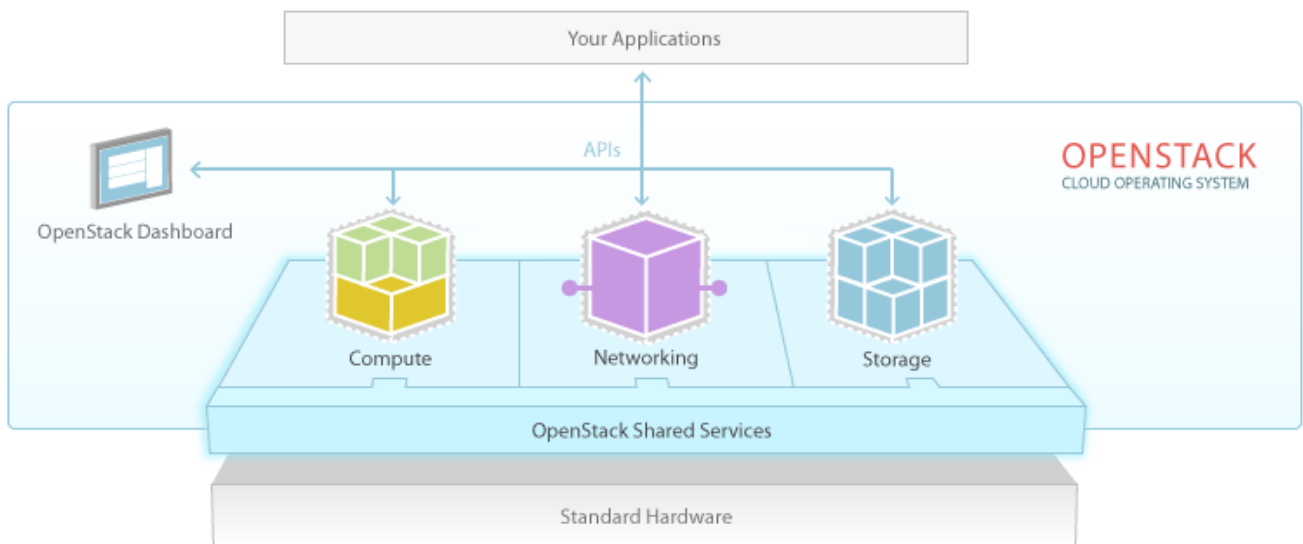


Ilustración 5 - Arquitectura de OpenStack

4.3.3. Proveedores Cloud Públicos

La utilización de cloud públicos se ha hecho muy popular en los últimos años. Esto es debido a las ventajas que aporta este tipo de IaaS. Una de ellas es el ahorro en gastos de infraestructuras. Puesto que no hace falta un lugar físico, ni máquinas, ni electricidad. El no depender de estos factores abarata mucho el coste final de la infraestructura, dado que se delegan al proveedor mediante un modelo de pago por uso, ajustando el gasto en infraestructura y obteniendo un ahorro muy grande en caso de que no se le vaya a dar un uso continuo a la infraestructura.

Algunos proveedores, además de proveer de IaaS, lo hacen de PaaS, ofreciendo entornos de desarrollo que se pueden combinar con el uso de infraestructuras ofrecidas por estos mismos.

Como ejemplos de este tipo de proveedores tenemos los siguientes:

Amazon AWS. El pionero en el campo del cloud computing. Dispone de una amplia cantidad de servicios, desde máquinas virtuales (EC2), almacenamiento (S3), bases de datos (RDS) y muchas más (En la ilustración 6 se pueden observar la gran cantidad de servicios).

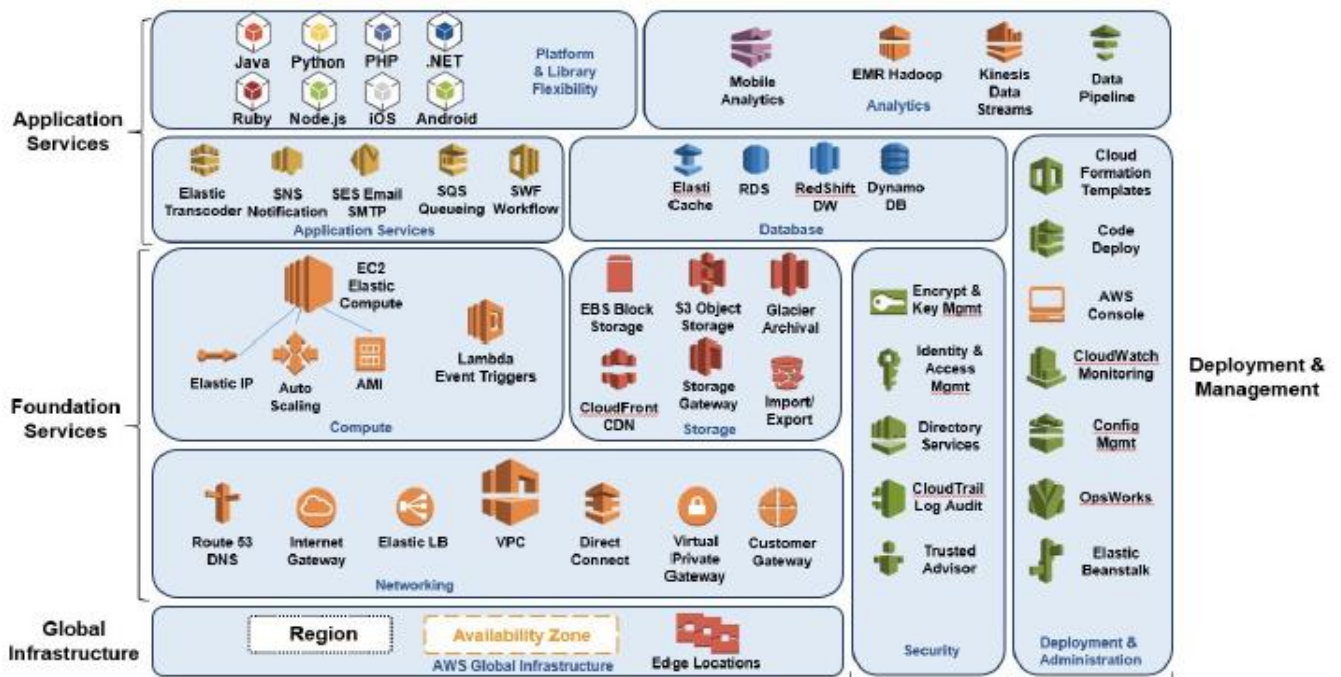


Ilustración 6 - Mapa de la arquitectura de servicios de AWS

La realización de pruebas de la plataforma en clouds públicos será en este proveedor debido a que la UPV dispone de acceso a este sin un coste añadido a través del programa AWS Educate, que proporciona créditos para dar soporte a actividades educativas.

- Microsoft Azure.** Microsoft es el desarrollador de otro gran proveedor de IaaS como es Azure (véase la ilustración 7 para observar la arquitectura de este servicio). Este compete gracias a que dispone de servidores en más regiones que cualquier otro proveedor. También dispone de una importante plataforma de SaaS como es Office 365 y otros productos de la marca, que están totalmente integrados y permiten un trabajo conjunto muy fácil.

Grid As A Service: Herramienta para el despliegue y gestión de un Grid en la nube para actividades educativas



Ilustración 7 - Arquitectura de Microsoft Windows Azure

- **Google Compute Engine.** También dispone de una plataforma en la que ofrece servicios de IaaS, así como de PaaS. Google ofrece múltiples herramientas para el trabajo con grandes cantidades de datos (BigData) a grandes velocidades, se pueden observar en el mapa conceptual de servicios de la nombrada plataforma en la ilustración 8. Es la plataforma más joven de las 3 que se presentan, pero cuenta con la experiencia en el campo de la computación de Google, algo que le ha ayudado a progresar en este mercado.

Cloud Platform Family

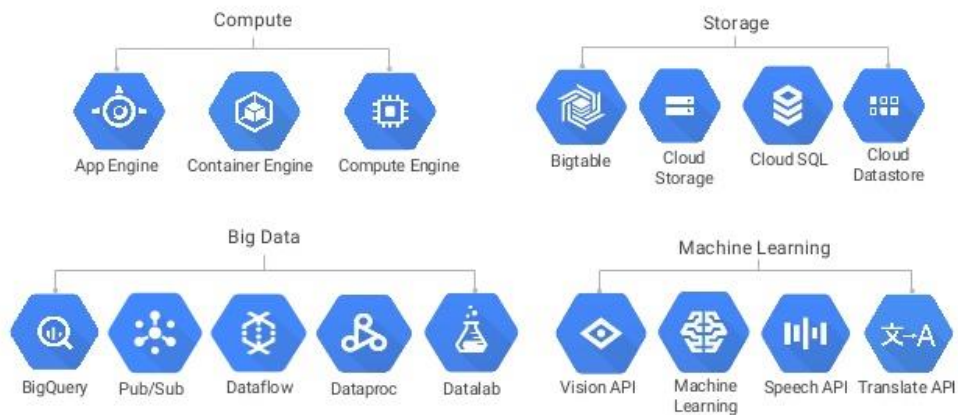


Ilustración 8 - Mapa de servicios de Google Compute Engine

5. Recurso Docente

En esta sección se va a desarrollar el recurso docente que se ha creado en este TFM, comenzando por el análisis de requisitos, tanto funcionales como no funcionales a través del estudio de un escenario de referencia. Después se procede a comentar la arquitectura que se ha utilizado a la hora de desarrollar esta herramienta. Y por último se desarrollará el flujo de trabajo de la misma, explicando paso a paso cómo funciona el recurso.

5.1. Análisis de requisitos

En este punto se va a desarrollar en primer lugar el escenario de referencia desde donde se extraerán los requisitos funcionales y no funcionales del recurso docente. A continuación se describe la arquitectura del recurso y su implementación.

5.1.1. Escenario de referencia

El escenario de referencia se enmarca en el Máster de Computación Paralela y Distribuida (MCPD) impartido por el Departamento de Sistemas Informáticos y Computación (DSIC) de la Universitat Politècnica de València. Concretamente en la asignatura Conceptos básicos de la computación Grid y Cloud (CCGC).

Grid As A Service: Herramienta para el despliegue y gestión de un Grid en la nube para actividades educativas

Esta asignatura se divide en cinco unidades temáticas en las que se desarrollan un conjunto de Actividades Educativas (A.E), que corresponden a las cinco unidades temáticas que conciernen al despliegue y uso de una IAG (ver tabla 3). En estas A.Es, los alumnos toman los roles tanto de usuario del Grid para la creación de aplicaciones Grid científicas, como administrador de recurso Grid para despliegue y configuración de una IAG desde cero.

Todas las A.Es se enmarcan dentro del contexto de un Aprendizaje Basado en Proyecto (ABP) o Project-based Learning (PBL), en la que todos los alumnos de forma conjunta crean una IAG utilizando el middleware proporcionado por la *toolkit* de Globus. En el ABP, cada alumno representa un dominio administrativo diferente perteneciente a la IAG que se va a desarrollar en el proyecto, en el que todos los alumnos configuran un recurso de cómputo y de datos en un mismo equipo (nodo Globus).

Cada dominio administrativo, dispone de una entidad certificadora (nodo CA) que se utiliza en el proyecto para certificar los nodos Globus como recursos Grid y a los alumnos como usuarios Grid de la IAG.

Para el desarrollo del ABP, a cada alumno se le proporciona un equipo (nodo Globus) con un sistema operativo Ubuntu 14.04. y la Globus Toolkit instalada. Los alumnos tienen privilegios de administrador para poder abordar las A.Es relativas a la configuración y administración de los recursos Grid.

Las A.Es según el rol (usuario Grid o administrador de recurso) que se realizan en las diferentes unidades temáticas del PBL se describen en la tabla 1.

De acuerdo a las A.Es identificadas en cada unidad temática, el recurso docente a desarrollar debe cumplir con los requisitos funcionales y no funcionales presentados en las siguientes secciones.

UNIDAD TEMÁTICA	ROL	A.E.	DESCRIPCIÓN
UT1. Seguridad Grid	Usuario	A.E.1	Generar certificados de usuario Grid a través de las CA de cada dominio.
	Administrador	A.E.2	Generar certificados de recurso Grid (hosts) a través de las CA de cada dominio.

		A.E.3	Configurar autenticación y autorización del recurso Grid.
UT2. Gestión de Trabajos Grid	Usuario	A.E.4	Definir trabajos Grid con <i>Resource Specification Language (RSL)</i> .
		A.E.5	CLIs para gestionar ciclo de vida de trabajos Grid.
	Administrador	E.A.6	Configuración y puesta en marcha del servicio GRAM del recurso Grid.
UT3. Gestión de Datos	Usuario	A.E.7	Definir y lanzar trabajos Grid que requieran de gestión de entrada/salida de datos.
		A.E.8	Definir y lanzar trabajos Grid que requieran de grandes volúmenes de datos.
	Administrador	A.E.9	Configuración y puesta en marcha del servicio GASS para el redireccionamiento de la Entrada/Salida y gestión del staging de los ficheros.
		A.E.10	Configuración y puesta en marcha del servicio GridFTP para la transferencia de ficheros de gran volumen.
UT4. Monitorización y Descubrimiento	Usuario	A.E.11	CLIs para consultar el sistema de Monitorización y descubrimiento Grid MDS.
	Administrador	A.E.12	Definir datos a publicar en el MDS de un recurso Grid.
UT5. Trabajo Final	Usuario / Administrador	A.E.13	Desarrollo de una aplicación Grid completa haciendo uso de los dos roles.

Tabla 3 - Unidades temáticas

5.1.2. Requisitos funcionales

La plataforma planteada debe cumplir los siguientes requisitos funcionales en base al escenario de referencia descrito en el punto anterior:

- **Debe facilitar el despliegue de una IAG en un proveedor cloud de forma agnóstica sin necesidad de acceder a la interfaz de este último.**

Para que ofrezca una mejora importante en comparación con lo que sería un despliegue manual, la plataforma debe permitir seleccionar diferentes proveedores

Grid As A Service: Herramienta para el despliegue y gestión de un Grid en la nube para actividades educativas

IaaS sin que el usuario deba de acceder a ésta en ningún momento. Solamente, indicando las credenciales válidas con las que poder desplegar en el proveedor.

- **Permitir la adición de nodos globus para su configuración como recurso Grid para la realización de las A.Es con rol de administrador.**

Se debe permitir desplegar en cualquier momento nodos con la toolkit de globus instalada y preparada para que el alumno los configura como un recurso Grid (cómputo o datos) de forma manual y así ejecutar las A.Es planeados con rol de administrador. Estos nodos deben de comunicarse con otros nodos dentro del mismo contexto de la IAG que se esté desarrollando en la ABP.

- **Permitir la adición de nodos completamente operativos a una IAG ya desplegada a través del recurso docente sin necesidad de configuración por parte del usuario.**

Una vez una IAG este de ya desplegada y operativa, el usuario debe poder desplegar uno o más nodos globus (recursos de cómputo, datos) o nodos CA junto con los ya existentes, y configurándolos automáticamente para que estén operativos y se integren con el resto de nodos de la IAG. Este requisito, tiene como objeto poder realizar aplicaciones Grid y medir la escalabilidad de estas en función del número de recursos disponibles.

- **Permitir la eliminación de nodos (globus o CA) de una IAG ya desplegada y sin necesidad de configuración por parte del usuario.**

Una vez una IAG este desplegada y operativa, el usuario debe poder eliminar uno o más nodos, reconfigurando automáticamente el resto de nodos de la IAG para que esta siga siendo operativa en su conjunto.

- **Eliminar una IAG y todos sus nodos de forma conjunta.**

En cualquier momento se le debe brindar al usuario la opción de finalizar y eliminar la IAG, liberando todos los nodos que la componen de una manera correcta y ordenada para no ocasionar errores y buscando el mínimo tiempo de repliegue.

5.1.3. Requisitos no funcionales

El recurso docente planteado, debe cumplir los siguientes requisitos no funcionales:

- **Debe ofrecer una interfaz usable para que el docente pueda interactuar con la IAG ofreciendo las funcionalidades descritas en la sección anterior**

Se debe ofrecer al usuario una interfaz por la que retroalimentar la información



relativa a la IAG desplegada y que permita desplegar y replegar los nodos correspondientes, además de interactuar con aquellos nodos que requieran de feedback por parte del profesor. (Un ejemplo de esto sería la inicialización de OpenCA que requiere de acciones por parte del profesor, dado que este indica el dominio administrativo que se quiere representar con la CA).

- **Debe ofrecer portabilidad en cuanto a proveedores IaaS.**

Una de las cosas que debe facilitar el recurso docente es el despliegue de una IAG en un cloud y para ello debe ser portable y funcionar en diferentes proveedores, tanto en ámbito privado (OpenStack, OpenNebula, etc.) como público (AWS, Azure, etc.).

- **Reducir el tiempo de despliegue y repliegue de una IAG en comparación con un despliegue manual.**

Este es uno de los objetivos del recurso docente, dado que lo que se busca es reducir el *overhead* técnico de los nodos.

- **Brindar escalabilidad a la infraestructura grid.**

La plataforma propuesta debe proporcionar al grid desplegado la capacidad de escalar horizontalmente bajo demanda del profesor, tanto escalado a más recursos, como a menos.

5.2. Arquitectura

El recurso docente tiene una arquitectura que parte de una máquina inicial desde donde el usuario despliega en el cloud un nodo que actúa como FrontEnd (FE) para el despliegue y repliegue de nodos en la IAG a través del Infraestructre Manager (IM). A partir de este momento, la máquina inicial ya no se encarga de nada, la orquestación la lleva a cabo el IM del nodo FE bajo la gestión del profesor.

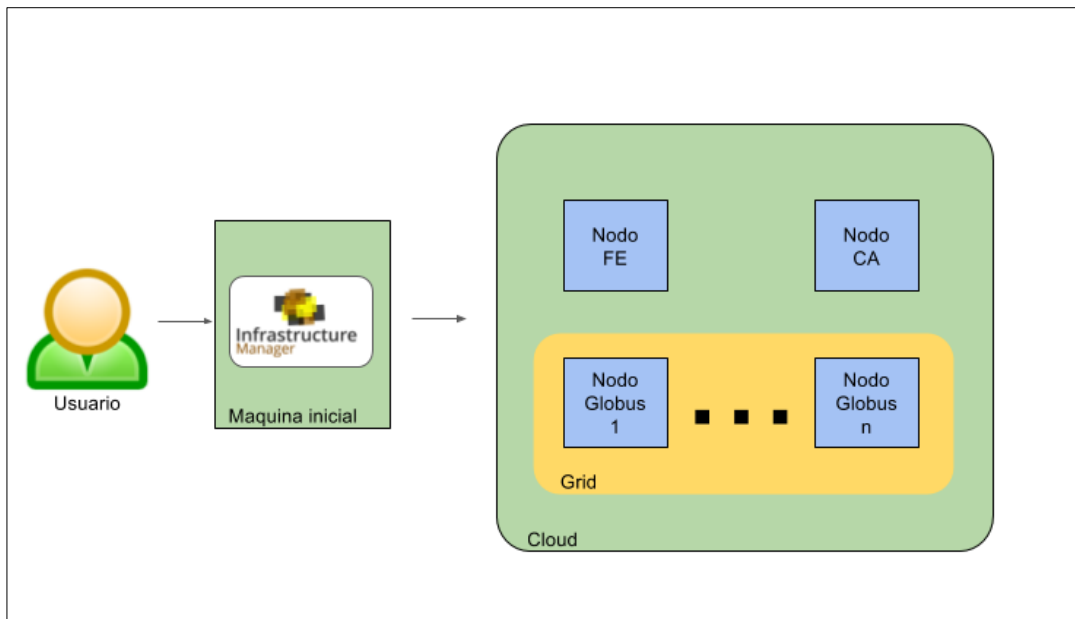


Ilustración 9 – Arquitectura inicial de la aplicación

Como se puede observar en la ilustración 1 el usuario solamente necesita disponer de una máquina física con el IM para poder desplegar el FE en un proveedor cloud, desde donde gestionará la IAG, desplegando y replegando nodos (globus o CA).

Desde el punto de vista de la comunicación entre los nodos de una IAG, en la Ilustración 2 podemos observar que el nodo FE es el nexo de unión entre todos ellos, puesto que gracias a él se pueden sincronizar los ficheros necesarios de los nodos Globus y CA. Además, este FE es el punto de acceso del profesor a la gestión de la IAG, ya que es el que le proporciona todas las operaciones referidas a los requisitos funcionales mediante un User Interface (UI) que proporciona una serie de Comand Lines interfaces (CLIs) para añadir o quitar nodos. El FE tiene acceso a todos los nodos de la IAG desde una red externa. La IAG posee una red interna en la que se encuentran todos los nodos globus, y aísla a todos estos nodos de la IAG desplegados en la nube, siendo esta accesible solo desde el FE.

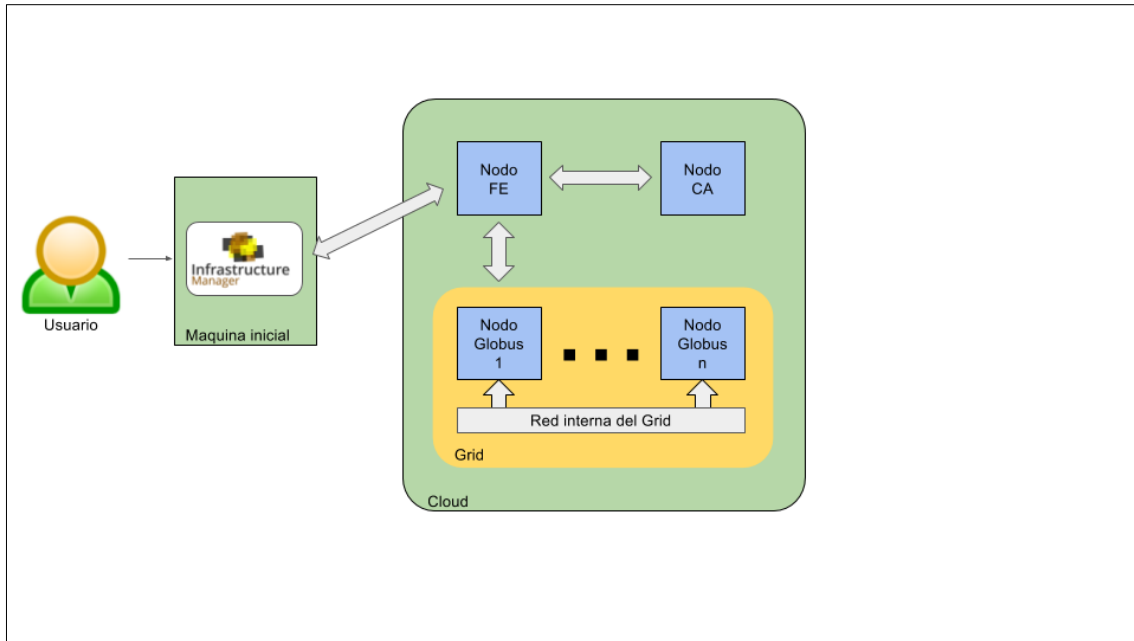


Ilustración 10 – Arquitectura de comunicaciones

Todos los nodos globus, CA y FE disponen de una interfaz de red con acceso a internet para poder comunicarse con sus usuarios, en el caso del FE el usuario es el profesor, en el caso de los nodos globus y CA los usuarios son los alumnos, pero cada uno tiene una configuración de sockets diferente para poder cumplir con sus requisitos. A continuación, se detallan las arquitecturas de servicios y sockets de cada nodo:

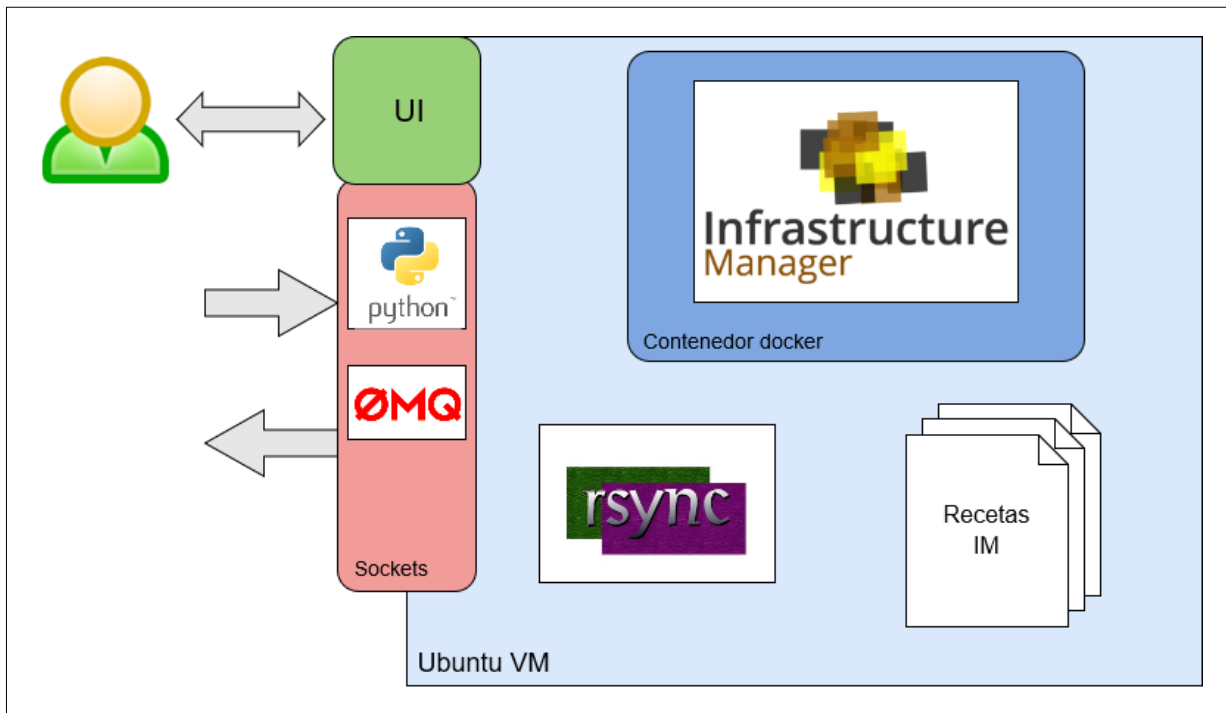


Ilustración 11 - Arquitectura de un nodo FE

- **Nodo FE:**

Este tipo de nodo es la pieza central de la IAG. Se conecta con todos los nodos y es el encargado de sincronizar todos los ficheros necesarios para hacer funcionar la IAG mediante el software RSync y los sockets establecidos.

También se encarga de gestionar los nodos mediante una instancia del IM levantada en un contenedor docker. A este IM se le aplican las diferentes recetas para los nodos que se quieran desplegar. Esto ayuda bastante a la hora de actualizar el sistema, puesto que puedes simplemente modificar las versiones de software de las recetas y si no hay ningún tipo de incompatibilidad en las actualizaciones, se pueden desplegar.

La estructura de conexiones del nodo consta de dos sockets:

- **Socket REQ (CA):** Un socket tipo REQ que conectará con el nodo CA y será el encargado de pedir credenciales al nodo CA, tanto para la inicialización de la IAG, como para la creación de nuevos nodos Globus.
- **Socket REQ (GLO):** Este socket conectará con cada uno de los nodos Globus cuando haya que inicializarlos. Se encargará de avisar a los nodos que las credenciales ya están transferidas y que todo está preparado para arrancar y utilizar los servicios de Globus.

- Nodo CA:

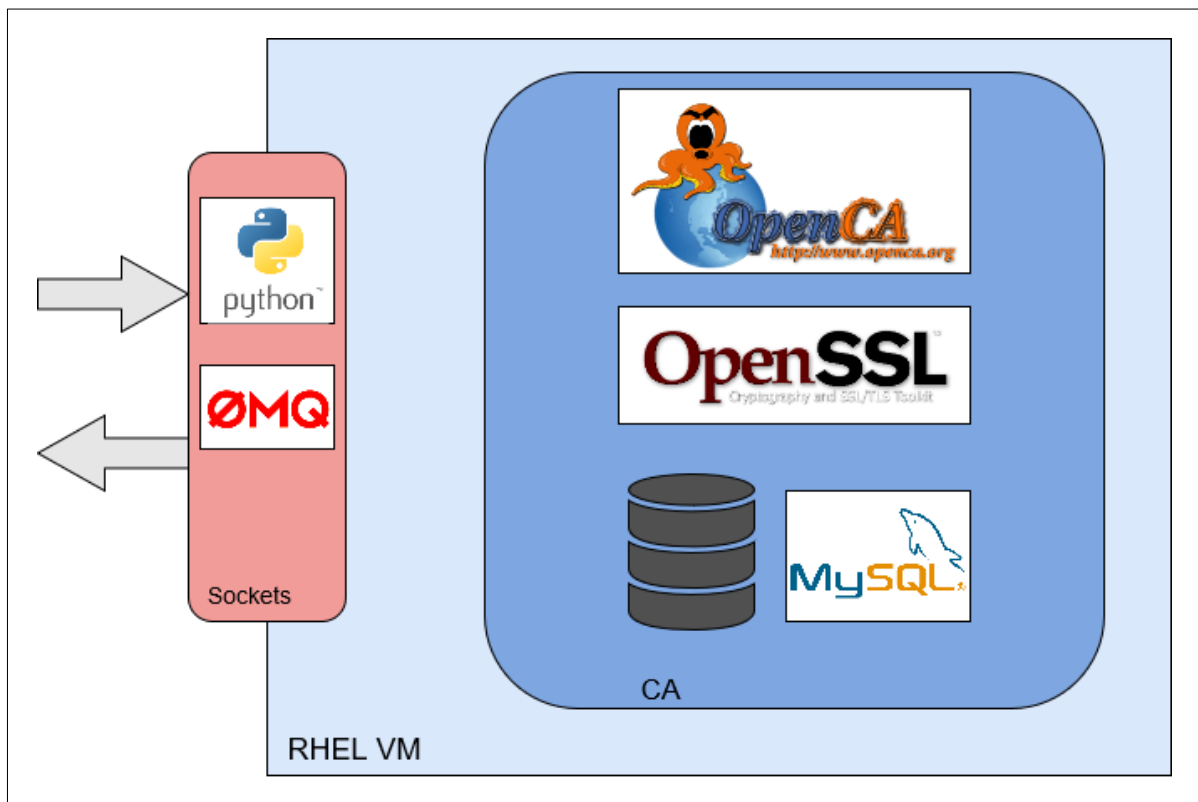


Ilustración 12 - Arquitectura de un nodo CA

Este nodo funcionará de autoridad certificadora para poder autorizar a los usuarios el acceso a la IAG y certificar los recursos de la misma. Para ellos se dispondrá de OpenCA, OpenSSL y una base de datos MySQL. Mediante estas herramientas el nodo será capaz de generar un certificado emulando ser una autoridad certificadora o CA. Con este certificado podrá firmar peticiones de certificados que generen los usuarios.

OpenCA también cuenta con una interfaz web, en la que el usuario debe inicializar la CA con los valores que desee. Los usuarios podrían registrar una petición de certificado y más adelante obtener el certificado. Pero por motivos de automatización y sencillez del sistema, la propia CA genera las credenciales, las peticiones de certificado, y las firma todo en el mismo nodo. De este modo podemos obtener credenciales para todos los nodos sin tener que requerir del acceso de todos los usuarios al nodo en cuestión.

La base de datos se utiliza para almacenar todos los datos de la CA, como el propio certificado de la CA y todas las peticiones y datos sobre estas que son lanzadas mediante OpenCA.

Grid As A Service: Herramienta para el despliegue y gestión de un Grid en la nube para actividades educativas

Las conexiones de este nodo permiten que se comunique con el nodo FE y que tenga una salida hacia la red por el puerto 80 para poder interactuar con la interfaz web.

Este nodo dispondrá de un socket tipo REP que recibirá del nodo FE las peticiones de creación de credenciales. De modo que estará siempre escuchando en un puerto para lanzar el script de creación de credenciales cuando sea necesario y en su caso devolverá un mensaje para confirmar el correcto funcionamiento.

- Nodo Globus:

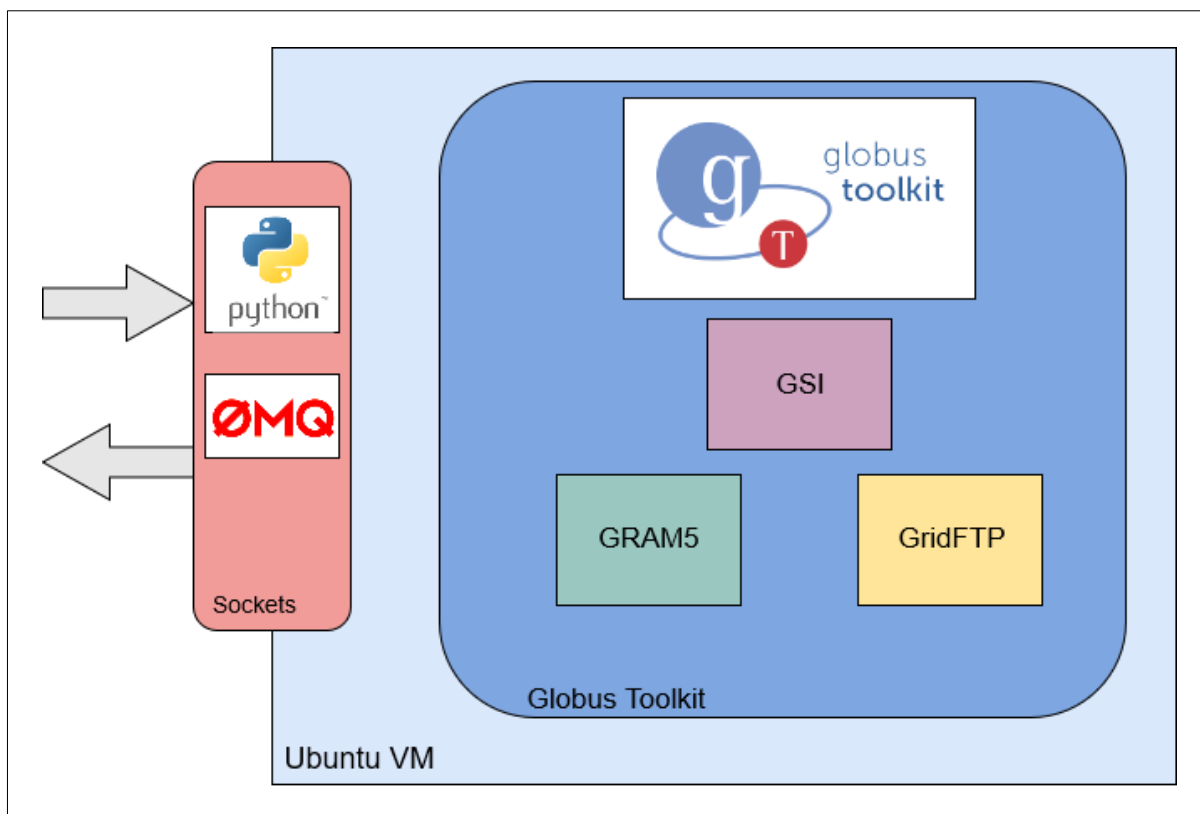


Ilustración 13 - Arquitectura de un nodo Globus

El nodo globus es el que actuará de recurso de cómputo o almacén de datos en la IAG. Al ser desplegado se configurará automáticamente y tendrá activos los servicios de Globus que permitirán a los usuarios trabajar con el nodo como parte de la IAG.

Para levantar todos los servicios necesitará de unas credenciales que firma la CA. Una vez tenga estas credenciales puede iniciar todos los servicios y funcionar correctamente.

Para poder funcionar junto al resto de nodos necesitará tener configurados varios ficheros con las direcciones y demás datos del resto de nodos globus. Esto será gracias al nodo

FE que sincronizará los ficheros necesarios con todos los nodos globos, para que todos se puedan ver en su red interna.

Este tipo de nodo consta de un socket de tipo REP que se utilizará para avisar a estos nodos de que los archivos necesarios para lanzar los servicios y funcionar correctamente ya están sincronizados, y de esa manera, ya pueden lanzar los procesos que correspondan.

5.3. Flujo de trabajo

Una vez se ha descrito la arquitectura desarrollada, a continuación se desarrolla el flujo de trabajo de la misma. Tomando como casos de uso, la inicialización de una infraestructura grid, la adición de nodos, la eliminación de nodos y la finalización de la infraestructura con su consiguiente limpieza de recursos. Para poder seguir mejor el flujo se han desarrollado unos mapas de flujo que se encuentran en cada sección (véase Ilustraciones 14-16).

5.3.1. Inicialización

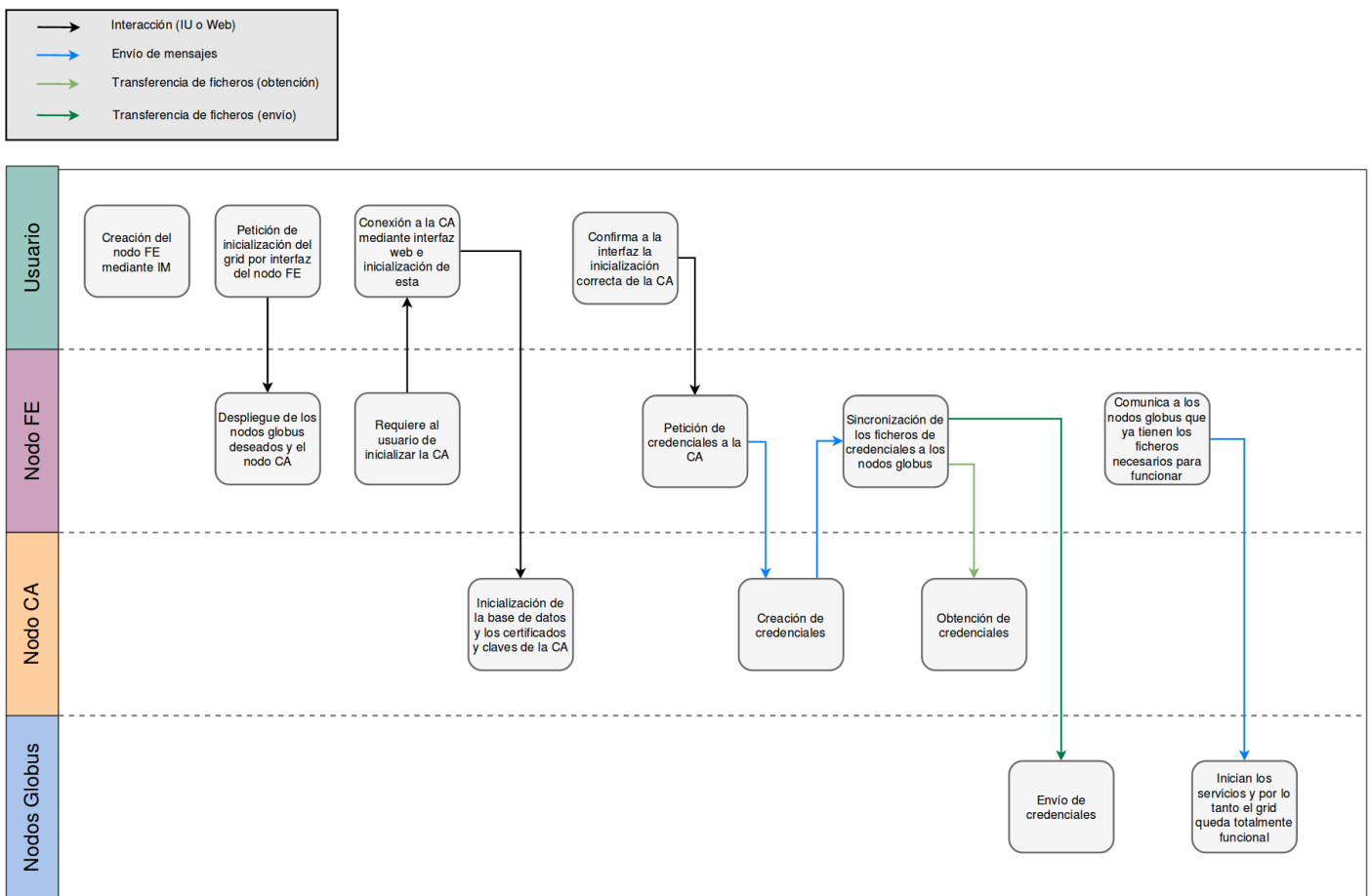


Ilustración 14 - Diagrama del flujo de la inicialización

El proceso de inicialización de una IAG consta de los siguientes pasos:

- *Levantamiento del nodo FE y acceso a la interfaz de este.* Gracias al IM y la receta en RADL del nodo FE, se lanza 1 nodo sobre el que se orquestrará el resto del despliegue. Mediante la interfaz del nodo FE el usuario podrá definir los parámetros del despliegue así como ver en cada momento qué operación se está llevando a cabo del despliegue. Para poder comenzar el docente debe haber definido unas credenciales de proveedor de infraestructuras sobre el que se lanzarán los nodos globus y CA.
- *El nodo FE levantará los nodos globus deseados y el nodo CA.* Mediante las recetas de dichos nodos y el IM, el nodo FE levantará los nodos inyectándose su clave SSH para poder utilizar el software de sincronización de ficheros. El número de nodos habrá sido definido en la interfaz por el docente previamente.
- *El usuario es requerido por la interfaz para inicializar la CA.* Para la inicialización de la CA se requiere de que el profesor acceda a la interfaz web de esta con los credenciales de administrador y complete varios pasos sencillos. Estos pasos, la dirección de la interfaz web y los credenciales son mostrados en pantalla por la interfaz. Y una vez terminado este paso el usuario confirmará que la inicialización ha terminado.
- *El nodo FE manda un mensaje pidiendo las credenciales al nodo CA previamente inicializado.* Tras la inicialización en el paso anterior, ahora la CA ya está lista para que el nodo FE le pida los credenciales necesarios para los nodos globus.
- *La CA recibe el mensaje, crea las credenciales y envía una respuesta al nodo FE.* Mediante los scripts necesarios, la CA genera y organiza todos los ficheros necesarios y luego responde al nodo FE.
- *El nodo FE prepara todos los ficheros y los sincroniza con los nodos globus involucrados.* El FE prepara primero los ficheros para que gestionen la red interna y luego recorre la lista de nodos globus y les va sincronizando a cada uno sus credenciales y los archivos necesarios para hacerlos operativos.
- *Una vez sincronizados, el nodo FE les comunica a los nodos globus que ya tienen los ficheros necesarios.* Mediante un mensaje les comunica a los nodos que ya tienen los ficheros y pueden iniciar todos los servicios.
- *Los nodos globus inician todos los servicios para tener la IAG operativa.* Tras el mensaje los nodos lanzan los servicios necesarios utilizando las credenciales proporcionadas y responden al nodo FE con el resultado de la inicialización.

5.3.2. Adición de nodos



La adición de nodos globus se llevará a cabo mediante la interfaz de usuario del nodo FE. El proceso consta de los siguientes pasos como se puede ver en la siguiente ilustración:

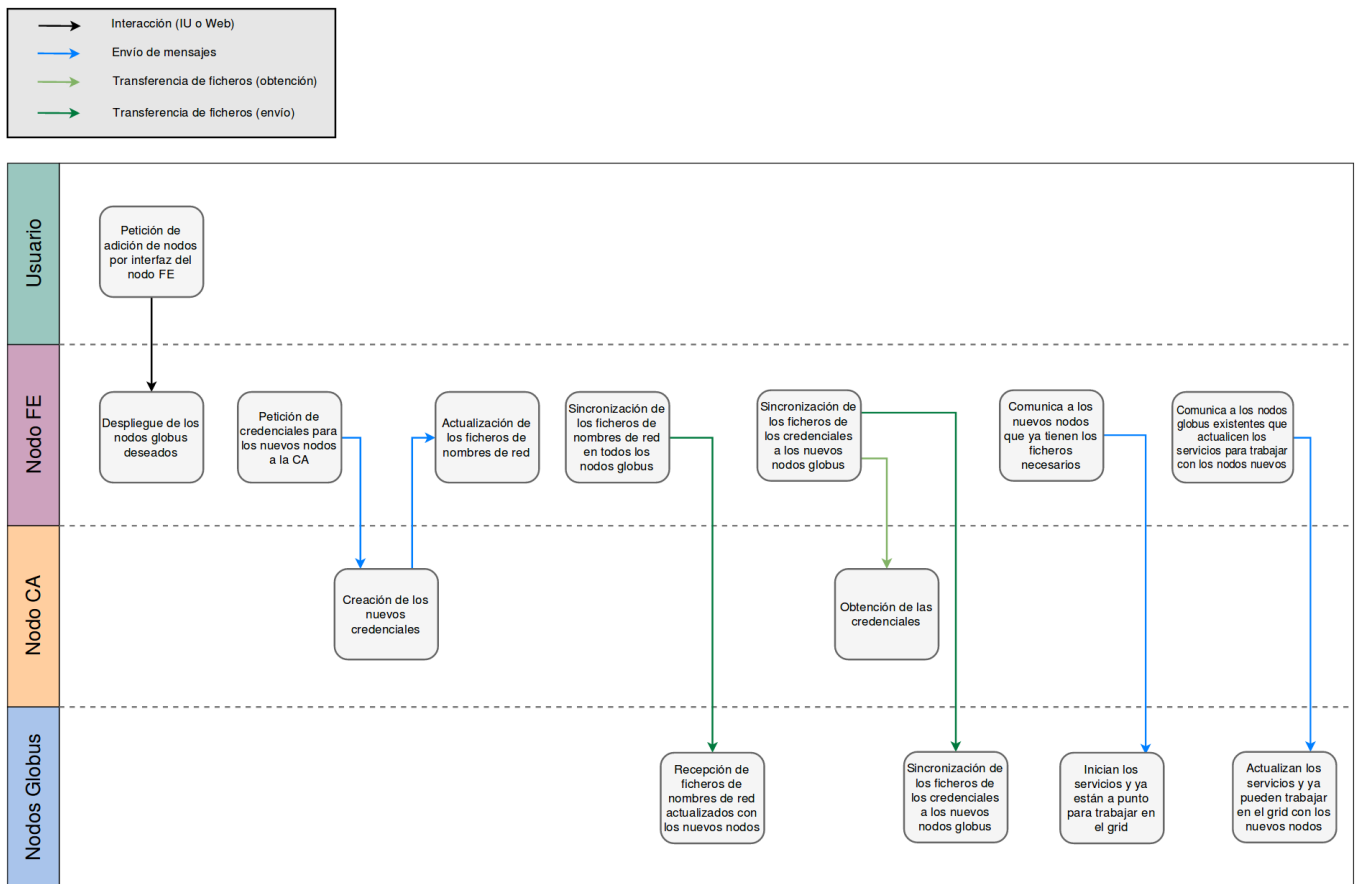


Ilustración 15 - Diagrama del flujo de la adición de nodos

- *El usuario mediante la interfaz especifica que quiere añadir uno o más nodos globus. Mediante la interfaz del nodo FE el usuario indica que quiere añadir más nodos al grid.*
- *El nodo FE despliega los nodos deseados mediante la receta correspondiente. Utilizando la receta de nodos globus, el nodo FE despliega los nodos necesarios a través del IM.*
- *Se le comunica a la CA que se quieren nuevas credenciales. Se envía un mensaje a la CA indicando el número de credenciales que se requieren, una por nodo.*
- *La CA recibe el mensaje y genera las credenciales. El nodo CA recibe el mensaje, genera las credenciales deseados, y luego devuelve la comunicación al nodo FE.*
- *Tras la respuesta de la CA el nodo FE actualiza los ficheros de gestión de nombres de red en todos los nodos. Para poder trabajar en red los nodos necesitan ficheros configurados con las direcciones de red y los nombres de todos los nodos. Estos ficheros los crea el nodo FE que tiene acceso a todos los datos de los nodos por el IM y luego los sincroniza.*

Grid As A Service: Herramienta para el despliegue y gestión de un Grid en la nube para actividades educativas

- *Sincronización de los ficheros de las credenciales.* Se sincronizan todos los ficheros que la CA había generado anteriormente en cada uno de los nodos añadidos.
- *Finaliza el proceso enviando un mensaje a los nodos Globus para avisar que ya están las credenciales sincronizadas.* Con el mensaje final, los nodos ya saben que pueden lanzar los servicios necesarios para poder tener el grid funcionando con los nodos añadidos correctamente.

5.3.3. Borrado de nodos

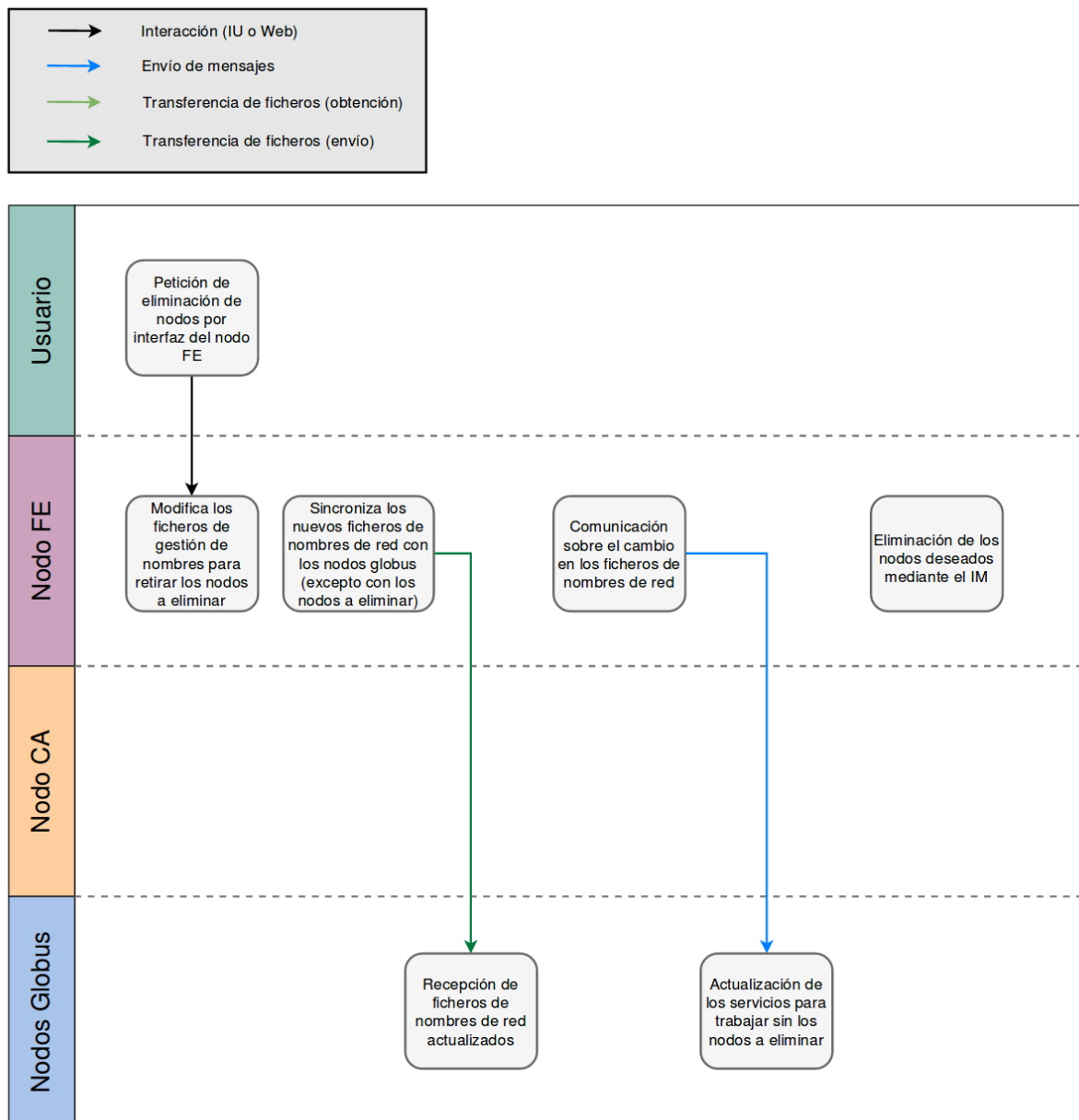


Ilustración 16 - Diagrama de flujo de la eliminación de nodos

Esta operación consta de las siguientes fases:

- *El usuario mediante la interfaz especifica que quiere borrar uno o más nodos. A través de la interfaz el usuario elige si quiere borrar un cierto número de nodos o especifica el nodo exacto que desea eliminar.*
- *Se modifican los ficheros de gestión de nombres de red para retirar las direcciones de los nodos a eliminar y se sincronizan. Se preparan los ficheros para que nadie pueda verlos más en la red y se sincronizan los ficheros en los nodos.*
- *El nodo FE avisa a los nodos que hay novedades en la red. Mediante un mensaje, el nodo FE avisa al resto (exceptuando los que van a ser eliminados) de que los ficheros de gestión de redes han cambiado.*
- *Tras esto el nodo FE procede a eliminar las máquinas correspondientes a los nodos. Mediante el IM el nodo se encargará de eliminar las máquinas deseadas y la IAG seguirá funcionando con los nodos restantes.*

5.3.4. Finalización

La finalización de una IAG es la operación más sencilla gracias a las facilidades que nos brindan las infraestructuras cloud. Puesto que todos los archivos de configuración son efímeros y deberán crearse en cada despliegue, la única fase de esta operación es la eliminación de todos los nodos por parte del nodo FE. Y después la eliminación del nodo FE por parte del profesor, aunque esto no es necesario para eliminar el grid, pero si para ahorrar en recursos y si no se prevé un uso próximo de la misma.



6. Pruebas y resultados

Para valorar la funcionalidad del recurso desarrollado se realizan diversas mediciones de tiempo en el despliegue de los diferentes tipos de nodos. De modo que se pueda apreciar el tiempo empleado y evaluar el ahorro de tiempo conseguido.

La realización de estas pruebas se ha llevado a cabo en una infraestructura cloud Open Nebula desplegada en el clúster *Ramses*¹⁰, perteneciente al grupo de *Grid y Computación de Altas Prestaciones* (GRyCAP) de la Universidad Politécnica de Valencia. Y los registros se han tomado utilizando los segundos como unidad de medida.

Para la evaluación de tiempos se han usado dos criterios, por un lado, evaluar el tiempo que tarda una máquina en desplegarse, esto parte desde que se lanza el despliegue de la máquina hasta que comienza la contextualización. Y por otro lado, se medirá el tiempo que se emplea en la contextualización de los diferentes nodos. Tiempo en el que se le aplican a la máquina todos los cambios necesarios para cumplir con su papel en la infraestructura (instalación de software, configuración de este, creación de usuarios, etc.).

Para poder observar de una manera más clara el impacto de cada contextualización en cada tipo de nodo. Se van a tomar tiempos de los diferentes nodos utilizados en el despliegue: nodo FrontEnd (FE), nodo de Entidad Certificadora (CA) y nodo de cómputo o nodo Globus (GLO). En la ilustración 17 podemos observar las diferencias de tiempos entre los diferentes tipos de nodos.

¹⁰ <http://www.grycap.upv.es/view.php/GRyCAP/Infraestructura/Equipos/ramses>

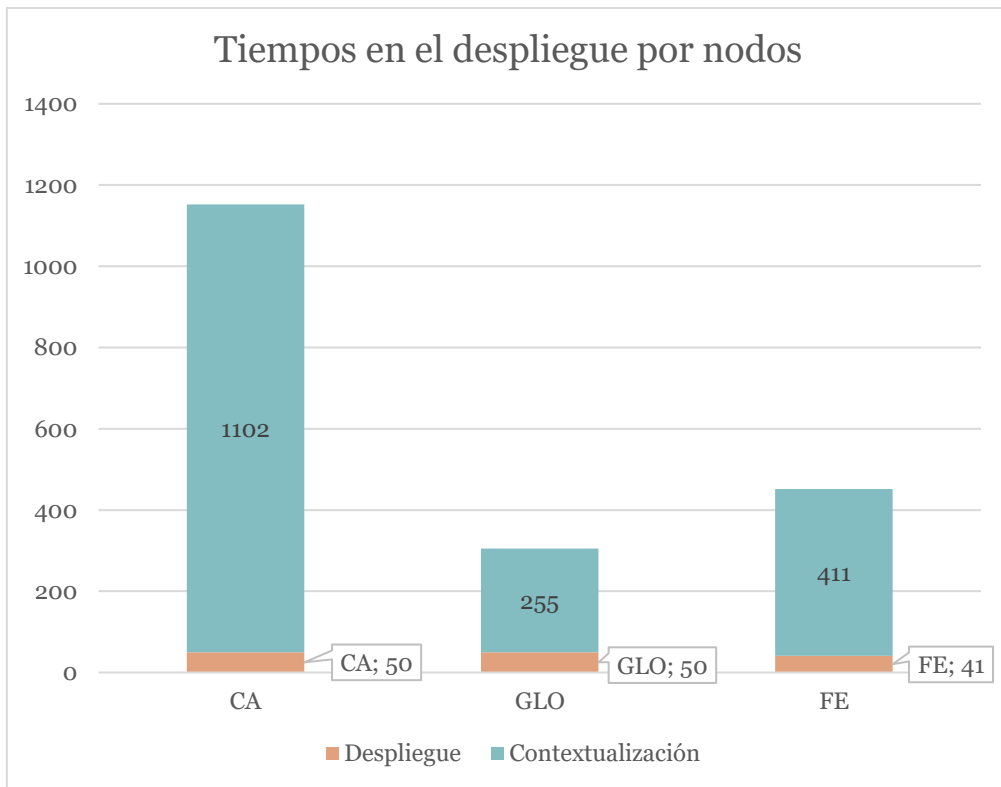


Ilustración 17 - Gráfica de tiempos de despliegue por nodos

Como se puede observar, el tiempo de despliegue en el que se levantan las máquinas es muy cercano y no varía prácticamente nada. Donde se aprecia una gran diferencia es en la contextualización de las máquinas. En ese aspecto se puede ver como el nodo CA requiere de mucho más tiempo. Esto es debido a que requiere de muchos más paquetes de software y muchos comandos de configuración. El nodo Globus y el nodo FE en cambio, tienen unos tiempos más reducidos. Esto es debido a que no requieren de la instalación de muchos paquetes, o como en el caso del FE, solo requiere de la descarga de muchos, pero pequeños ficheros.

Los tiempos son mucho mayores si en lugar de utilizar el recurso docente se hubiera tenido que desplegar la máquina y contextualizar cada nodo a mano uno a uno. Una medición de este tipo de despliegue no es requerida debido a su falta de precisión puesto que las tareas se realizan por el usuario y no es algo automático.

En cuanto a la evaluación de casos de uso, podemos observar en la ilustración 18 el tiempo empleado en desplegar y dejar funcional una infraestructura grid completa. No se ha medido el tiempo que se tardaría en crear la infraestructura sin este recurso, pero observando

los tiempos obtenidos no hace falta para ver que se cumple el objetivo de desplegar un grid de una manera casi automática y ahorrando mucho tiempo que el docente podrá invertir en otra tarea.

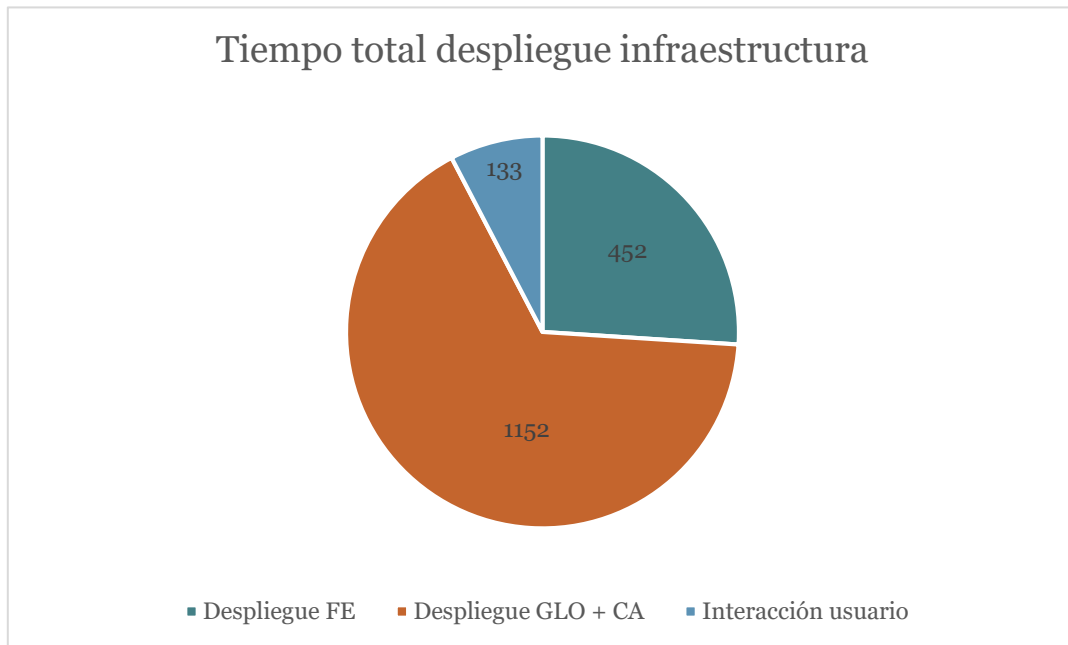


Ilustración 18 - Gráfico de tiempo total del despliegue de una infraestructura

Se puede observar que el grueso del despliegue es empleado en los nodos Globus y CA, esto es debido a que la CA como se podía observar en la ilustración 17 es un nodo que cuesta de desplegar, y al desplegarse en paralelo, este hace de cuello de botella en cuanto a coste temporal se refiere. La parte más reducida es la interacción del usuario que también incluye las partes de configuración que realiza la interfaz. Puesto que estas son transacciones sencillas (con casos como por ejemplo el envío del fichero hosts) y la interacción con el usuario es también bastante rápida, se aprecia que no llega a emplear ni un 25% del tiempo total.

Aun habiendo medido tiempos, hay que recalcar que una parte de la configuración depende de lo que tarde el usuario en manejar la interfaz y en inicializar la CA, por lo que el tiempo nunca será el mismo, pero como conclusión se puede observar el ahorro de tiempo gracias al recurso docente desarrollado.

7. Trabajos futuros

Una vez desarrollada la plataforma surgen diversas ideas por las que seguir desarrollando y diversos puntos en los que mejorar el recurso ideado. Posteriormente a la entrega de este trabajo seguiré desarrollando el recurso por lo que algunos de estos puntos estarán desarrollados en un futuro.

Por una parte la interfaz se podría llevar a otro nivel con un desarrollo en web utilizando tecnologías como html5 y AngularJS, este es un punto importante, aunque no se requiere de una interfaz de este tipo, podría ser algo muy positivo. De este modo podría crearse una interfaz más visual, más agradable para el usuario que no está familiarizado con el uso de consolas de línea de comandos. Esto sería de utilidad para poder observar el estado del despliegue completo mediante algún tipo de mapa gráfico y poder hacerse una idea rápidamente de si el despliegue va correctamente o no.

Por otro lado, una posible mejora del recurso docente sería aumentar el grado de personalización de la infraestructura desplegada añadiendo múltiples parámetros configurables por el usuario. Esto brindaría al docente que utilice la herramienta de la capacidad de personalizar el grid con parámetros que podrían ir desde simples configuraciones como puertos de sockets o nombres de red personalizables, hasta dar una opción a personalizar ligeramente el contenido de los nodos desplegados con diferentes software o herramientas que se añadirían en tiempo de despliegue.



8. Conclusiones

El recurso docente diseñado e implementado en este trabajo permite crear en la nube infraestructuras Grid virtuales y ofrecerlas como un servicio, en la que los docentes son capaces de automatizar el despliegue de este tipo de infraestructuras y escalarla según las necesidades del número de alumnos. Dado que es posible desplegar este tipo de infraestructura sobre proveedores de Cloud público, el recurso docente es ideal para cursos en los que el número de alumnos obligue a escalar la infraestructura a dimensiones no abordables por las infraestructuras Cloud on-premise disponibles en la institución educativa.

El uso de herramientas de configuración automatizada de infraestructuras virtuales facilita el proceso de despliegue de las mismas y permite obtener entornos de ejecución reproducibles en cuestión de minutos lo que permite garantizar la consistencia de las actividades educativas a lo largo de diferentes ediciones de la misma asignatura. Esto permite reducir la carga de trabajo del profesor, que puede dedicar ese tiempo a mejorar las actividades educativas sin tener que dedicar un tiempo valioso al proceso de despliegue y configuración de infraestructuras computacionales complejas como es el caso de una infraestructura Grid.

9. Bibliografía

- [1] Salvator Roberto Amendolia, Florida Estrella, Waseem Hassan, Tamas Hauer, David Manset, Richard McClatchey, Dmitry Rogulin, and Tony Solomonides. MammoGrid: A Service Oriented Architecture Based Medical Grid Application. pages 939–942. Springer Berlin Heidelberg, 2004.
- [2] G. Andronico, V. Ardizzone, R. Barbera, R. Catania, A. Carrieri, A. Falzone, E. Giorgio, G. La Rocca, S. Monforte, M. Pappalardo, G. Passaro, and G. Platania. 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*, pages 304–305. IEEE.
- [3] Adam L. Bazinet, Daniel S. Myers, John Fuetsch, and Michael 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):517–522, 2007.
- [4] Miguel Caballer, Ignacio Blanquer, Germán Moltó, and Carlos de Alfonso. Dynamic management of virtual infrastructures. *Journal of Grid Computing*, 13(1):53–70, 2015.
- [5] Ann Chervenak, Ian Foster, Carl Kesselman, Charles Salisbury, and Steven 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):187–200, 2000.
- [6] Dietmar W. Erwin. UNICORE: A Grid computing environment. *Concurrency and Computation: Practice and Experience*, 14(13-15):1395–1410, nov 2002.
- [7] A. Forti, S.R. Bavikadi, C. Bigongiari, G. Cabras, A. De Angelis, B. De Lotto, M. Frailis, M. Hardt, H. Kornmayer, M. Kunze, and M. Piraccini. Grid services for the magic experiment. In *Frontiers of Fundamental Physics*, pages 333–337. Springer Netherlands, Dordrecht, 2006.
- [8] I. Foster, Carl Kesselman, and Steven Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications*, 15(3):200–222, aug 2001.
- [9] José A. González-Martínez, Miguel L. Bote-Lorenzo, Eduardo Gómez-Sánchez, and Rafael Cano-Parra. Cloud computing and education: A state-of-the-art survey. *Computers & Education*, 80:132–151, 2015.
- [10] Attila Kertész and Péter Kacsuk. *A Taxonomy of Grid Resource Brokers*, pages 201–210. Springer US, Boston, MA, 2007.
- [11] D. Kranzlmüller, J. Marco de Lucas, and P. Öster. The European Grid Initiative (EGI). In *Remote Instrumentation and Virtual Laboratories*, pages 61–66. Springer US, Boston, MA, 2010.
- [12] Johan Montagnat, Ákos Frohner, Daniel Jouvenot, Christophe Pera, Peter Kunszt, Birger Koblitz, Nuno Santos, Charles Loomis, Romain Texier, Diane Lingrand, Patrick Guio, Ricardo Brito Da Rocha, Antonio Sobreira de Almeida, and Zoltán Farkas. A Secure

Grid Medical Data Manager Interfaced to the gLite Middleware. *Journal of Grid Computing*, 6(1):45–59, mar 2008.

[13] Nancy Wilkins-Diehr. Special Issue: Science Gateways—Common Community Interfaces to Grid Resources. *Concurrency and Computation: Practice and Experience*, 19(6):743–749, apr 2007

[14] Ellert, M., Grønager, M., Konstantinov, A., Kónya, B., Lindemann, J., Livenson, I., J.L. Nielsen, M. Niinimäki, O. Smirnova & Wäänänen, A. (2007). Advanced Resource Connector middleware for lightweight computational Grids. *Future Generation computer systems*, 23(2), 219-240.

[15] Segrelles, J. D., & Moltó, G. (2016, October). Assessment of cloud-based Computational Environments for higher education. In *Frontiers in Education Conference (FIE)*, 2016 IEEE (pp. 1-9). IEEE.

[16] Segrelles, J. Damian; Molto, German; Caballer, M. (2015) Remote Computational Labs for Educational Activities via a Cloud Computing Platform. In *Proceedings of the Information Systems Education Conference (ISECON)*, pages 309-321, Orlando, 2015.

[17] Erwin, D. W., & Snelling, D. F. (2001, August). UNICORE: A Grid computing environment. In *European Conference on Parallel Processing* (pp. 825-834). Springer Berlin Heidelberg.