

**Departamento de Arquitectura de Computadores y
Automática
Facultad de Informática
Universidad Complutense de Madrid**



Proyecto De Sistemas Informáticos
**SmartCloud: Gestión Eficiente de Recursos
Computacionales de la Administración Pública**

Autores:

Ailyn Baltá Camejo
César Cayo Ventura
Javier Bachrachas Peterburg

Tutores:

José Antonio Martín H.
José Luis Vázquez-Poletti

Madrid, Junio de 2013



SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

Los alumnos abajo firmantes autorizan a la Universidad Complutense de Madrid a difundir y utilizar solo con fines académicos, no comerciales, mencionando expresamente a sus autores, tanto la propia memoria, como el código, los contenidos audiovisuales incluso si incluyen imágenes de los autores, la documentación y/o el prototipo desarrollado.

En Madrid, a 21 de Junio de 2013.

Javier Bachrachas Peterburg

Ailyn Baltá Camejo

César Cayo Ventura





AGRADECIMIENTOS

En primer lugar quisiéramos agradecer a nuestros tutores José Luis Vázquez Poletti y José Antonio Martín H. por la guía y ayuda durante el desarrollo de este proyecto. A su vez, queremos agradecer a toda la comunidad iPython.

A mis seres queridos que están lejos pero siempre presentes. A Javier, Analay y Antonio por estar a mi lado, apoyándome y pendientes. A mis amigos, por brindarme su amistad y cariño. A todos por creer en mí y darme fuerzas para continuar.

Ailyn

A mi familia que siempre me han acompañado tanto de forma directa como indirecta, a mis amigos por sus buenos consejos en situaciones difíciles, y a todas las personas que han hecho esto posible.

César

A mis padres, a mi hermana Janine, a mis amigos de España que son mis compañeros de batalla, a mis amigos del otro lado del Atlántico que siempre están presentes sin estarlo, y a toda la gente random que ha contribuido sin saberlo.

Javier

A TODOS, MUCHAS GRACIAS.





TABLA DE CONTENIDOS

Agradecimientos	5
Tabla De Contenidos	7
1. Resumen	11
2. Abstract.....	13
3. Introducción	15
4. Estado Del Arte	19
4.1. Cloud-Computing	19
4.1.1. Beneficios y Desventajas	19
4.1.2. Tipos De Servicios	20
4.1.2.1. Infraestructura Como Servicio (IAAS).....	20
4.1.2.2. Plataforma Como Servicio (PAAS)	21
4.1.2.3. Software como servicio (SAAS).....	21
4.1.3. Tipos de Infraestructuras	21
4.1.3.1. Privadas.....	22
4.1.3.2. Públicas	22
4.1.3.3. Híbridas	22
4.2. Virtualización.....	23
4.2.1. Hipervisores	23
4.2.1.1. Hipervisor tipo I.....	23
4.2.1.2. Hipervisor tipo II	24
4.2.1.3. Hipervisor Híbrido.....	24
4.3. Valoración De Aplicaciones Existentes	25
4.3.1. Conclusiones Del Estudio.....	25
4.4. Nuevas Estrategias y Soluciones	25
4.4.1. Hipervisores: VMware y VirtualBox.....	25
4.4.2. Gestores PBS.....	26
4.4.3. Herramientas De Cálculo	26





5.	Arquitectura Del Sistema	29
5.1.	Aproximación de alto nivel	29
5.2.	Arquitectura Por Capas	29
5.3.	Implementación De Las Capas.....	31
5.3.1.	Capa Inferior: Nodo De Procesamiento	31
5.3.2.	Capa Intermedia: PBS Torque y NFS	32
5.3.2.1.	Gestor PBS Torque	32
5.3.2.1.1.	Características más importantes del sistema PBS	32
5.3.2.1.2.	Importancia de un Gestor PBS.....	33
5.3.2.2.	Network File System (NFS).....	33
5.3.2.2.1.	Características	34
5.3.2.3.	Detalles de implantación de PBS en SmartCloud	34
5.3.3.	Capa Superior: Frontend y Herramienta de Cálculo iPython	35
5.3.3.1.	Frontend	35
5.3.3.2.	iPython.....	36
5.3.3.2.1.	Notebook	37
5.3.3.3.	Detalles de implantación de Notebook en SmartCloud.....	38
6.	Instalación y Configuración de los componentes internos de la solución.....	41
6.1.	Gestor PBS Torque	41
6.1.1.	Nodos de Cómputo	42
6.1.2.	Para Crear tpackages	42
6.1.3.	Establecer Torque como servicio (Opcional)	43
6.1.4.	Configuración Básica	44
6.1.5.	Iniciar/Configurar TORQUE en el servidor (pbs_server)	44
6.1.6.	./torque.setup	45
6.1.7.	Especificar nodos de cómputo	46
6.1.8.	Configurar TORQUE en los nodos de cómputo	48
6.1.9.	Finalizar configuraciones	48
6.1.10.	Configuración inicial del servidor.....	48
6.1.11.	Cancelar trabajos.....	49
6.1.12.	Configuración de colas	50
6.2.	Secure SHell(SSH) & autenticación RSA	51
6.2.1.	Identificación SSH mediante clave	51
6.3.	Network File System (NFS).....	53
6.3.1.	Notas preliminares.....	53





6.3.2.	Instalando NFS.....	54
6.3.3.	Exportando directorios en el servidor	55
6.4.	iPython	58
6.5.	Apache Web Server & mySql.....	59
6.5.1.	Instalacion y configuracion de Apache 2.....	59
6.5.2.	Instalar MySQL.....	61
6.5.3.	Instalar PHP	63
7.	Estudio De Costes e Impacto Económico.....	67
8.	Casos De Uso.....	71
8.1.	Registro en SmartCloud	71
8.2.	Login con Google Id.....	72
8.3.	Login sin Google Id.....	73
8.4.	Utilizar iPython para generar/utilizar notebook.....	73
8.5.	Utilizar el cluster (iniciar y detener)	74
8.6.	Sincronizar archivos con Google Drive	75
8.7.	Consultar estado del clúster	76
9.	Interfaz del Sistema	77
9.1.	Página de Inicio.....	77
9.2.	Formulario de Registro.....	78
9.3.	Notebook	78
9.3.1.	Listados de Notebooks	78
9.3.2.	Clúster	79
9.3.3.	Google Drive	79
9.3.4.	Información del clúster	80
9.3.5.	Notebook	80
10.	Contribuciones y Trabajos Futuros	81
11.	Repercusión	83
12.	Anexos.....	85
12.1.	Guías De Instalación y Configuración	85
12.1.1.	Consideraciones Generales.....	85
12.1.2.	Virtualización del Nodo.....	85
12.1.2.1.	Establecer control de arranque y apagado del nodo	89
12.1.2.2.	Añadir Tarea Arrancar máquina virtual.....	90
12.1.2.3.	Añadir Tarea Apagar máquina virtual	92
12.1.3.	Configuración	94





SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

12.1.3.1.	Dentro de la imagen Server.....	94
12.1.3.1.1.	Configurar los nodos para ser accesibles por el servidor.....	94
12.1.3.1.2.	Configurar los nodos para ser accesibles por pbs_server	94
12.1.3.1.3.	Scripts iniciales.....	95
12.1.3.2.	Dentro de la imagen Nodo	95
12.1.3.2.1.	Configurar servidor dentro de los nodos.....	95
12.1.3.2.2.	Configurar el nombre de cada nodo	96
12.1.3.2.3.	Scripts iniciales.....	96
12.2.	Manual De Usuario	97
12.2.1.	Frontend	97
12.2.1.1.	Registro En SmartCloud.	97
12.2.1.2.	Login con Google ID	100
12.2.1.3.	Login sin Google Id.....	102
12.2.2.	Notebook	105
12.2.2.1.	Crear un notebook.....	105
12.2.2.2.	Abrir un notebook.....	105
12.2.2.3.	Configurar recursos del clúster	107
12.2.2.4.	Detener el clúster	108
12.2.2.5.	Sincronizar archivos en Google Drive	109
12.2.2.6.	Consultar el estado del clúster	110
12.3.	Artículos de Referencia y Publicaciones	112
12.3.1.	Artículo en HPC In the Cloud	112
12.3.2.	Artículo en Tribuna Complutense	113
12.3.3.	Artículo en la Facultad de Informática, UCM	114
12.3.4.	Publicación para el Congreso Español de Informática	115
13.	Glosario	121
14.	Referencias	125





1. RESUMEN

SmartCloud es una solución completa (IaaS + SaaS) siguiendo el paradigma de cloud computing. A través de una arquitectura de capas escalable se implementa una nube privada que gestiona los elementos de una infraestructura ya existente añadiendo a los mismos la funcionalidad de nodo de procesamiento en los tiempos de inactividad sin afectar su utilización habitual.

Sobre esta plataforma se ha implementado una solución de cálculo científico open source (iPython) que utiliza los recursos antes mencionados como soporte hardware para procesamiento paralelo y funcionalidades de clúster.

Como resultado de esta solución se puede brindar a los usuarios un nuevo servicio sin inversión en hardware ni licencias con el valor añadido de la reutilización de infraestructuras ya existentes.

Palabras clave: Cloud Computing, iPython, reutilización hardware, optimización infraestructuras, educación.





2. ABSTRACT

SmartCloud is a full solution (IaaS + SaaS) following the Cloud Computing Paradigm. Through a scalable layer architecture, we have deployed a private cloud that manages existing infrastructure elements, adding new functionalities to them. These elements now work as processing nodes in their inactivity periods without affecting their regular usage.

Over this platform it has been deployed an Open Source scientific computing application (iPython) that uses the former mentioned resources for parallel processing and cluster functionalities.

As the result of this project, a new service can be provided to users with no investment in hardware or in software licenses with the added value of the reuse of existing infrastructures.

Keywords: Cloud Computing, iPython, hardware reuse, infrastructure optimization, education.





3. INTRODUCCIÓN

La situación económica actual ha hecho que muchos grupos de trabajo en informática enfoquen sus esfuerzos en encontrar nuevas maneras de optimizar recursos o ampliar los servicios brindados a los usuarios sin que esto implique un aumento en los costes [8].

El primer enfoque que surge ante esta necesidad es migrar los sistemas hacia el software de código abierto, algo que es tendencia actual en organizaciones tanto privadas como estatales [4]. Un ejemplo de esto es la decisión del gobierno de Extremadura de migrar sus sistemas a código abierto y Linux, con un ahorro estimado de 30 millones de euros anuales [3].

En términos de la capa de aplicación, en el mundo de la investigación y la enseñanza también existen alternativas de código abierto a herramientas bien establecidas en el mercado como Matlab o Mathematica, las que al ser un software propietario tienen coste de licencia y su desarrollo está cerrado a una compañía, impidiendo las mejoras de la comunidad. En este proyecto hemos realizado una implantación de iPython [1] como alternativa a las aplicaciones antes mencionadas, permitiendo lograr una funcionalidad similar en términos de potencia de cálculo, pero adaptada a las necesidades y paradigma de este proyecto.

Un paso más en el análisis del problema, nos lleva a plantear la pregunta de cómo optimizar el soporte hardware de las aplicaciones. Nuestra aproximación a este problema es la reutilización de infraestructura ya existente para aprovechar su capacidad de procesamiento en los tiempos de inactividad [6].

Tomando como ejemplo la situación de la Facultad de Informática de la Universidad Complutense de Madrid, existen grupos de investigación que necesitan realizar procesamiento de datos y como la Facultad no puede brindarle servicios de procesamiento (o bien por qué no los tiene, o por restricciones presupuestarias), deben recurrir a proveedores externos como por ejemplo Amazon.

Nuestra solución a este problema es utilizar los ordenadores de los laboratorios de la Facultad cuando no están siendo utilizados, y sobre cada uno de ellos





SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

iniciar una máquina virtual con un nodo de procesamiento, lo que tiene un impacto nulo en la utilización rutinaria de los recursos y nos permite generar un cluster dinámico de procesamiento [9].

Hay que recordar los problemas que conlleva utilizar proveedores externos para el procesamiento de datos en términos de latencia y de protección de datos[21], por lo que hemos creado la solución "SmartCloud" como una propuesta para optimizar recursos y lograr brindar un mejor servicio a usuarios tal y como se puede apreciar en la Figura 1.

La solución se trata de una plataforma de Cloud Computing de nube privada2 pues se lleva a cabo con los propios recursos existentes (máquinas, redes, almacenamiento, instalaciones) y su acceso está restringido a una red cerrada. Incluye un componente de Infrastructure as a Service (IaaS) y un componente de Software as a Service (SaaS) unidas por una capa de integración.

Se trata de una solución escalable y flexible, dado que al contar con un diseño arquitectónico por capas, se puede reutilizar o modificar la plataforma para implementar otras soluciones software sobre la misma.

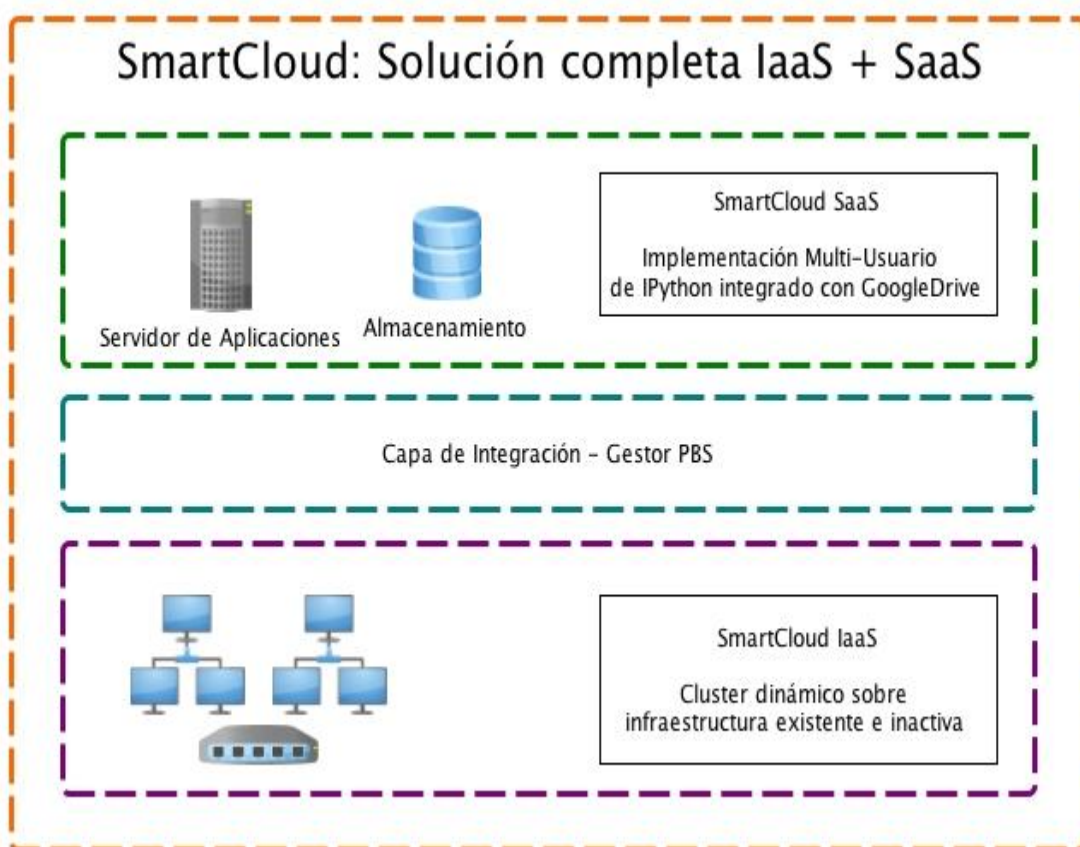


Ilustración 1 Diagrama de Solución SmartCloud integrando IaaS + SaaS

El servicio brindado al usuario final se trata de una implementación de iPython multiusuario, que permite que cualquier usuario del sistema acceda a través de





SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

un navegador de internet a nuestra aplicación, permitiéndole realizar simulaciones, gráficos, cálculos, procesamiento de código Python y otras funcionalidades con la ventaja del procesamiento distribuido. El procesamiento no se realiza en el ordenador del cliente sino en nuestro servidor centralizado y en el cluster dinámico antes mencionado.

Nuestra plataforma permite adicionalmente la sincronización de archivos de usuario con Google Drive, permitiendo el trabajo colaborativo.

En esta contribución primero se estudiará el estado del arte para evaluar las alternativas existentes. Luego, se explicará la arquitectura de la solución haciendo la distinción entre diseño de alto y bajo nivel. Posteriormente se detallará el modelo de ahorro de costes e impacto económico de la solución, lo que ayuda a entender la motivación del proyecto. Para terminar se explicarán las conclusiones del trabajo y las posibles ampliaciones que puede tener.





4. ESTADO DEL ARTE

4.1. CLOUD-COMPUTING

El Cloud Computing se ha convertido en la tecnología más demandada por las empresas gracias a las ventajas que ofrece sobre otro tipo de tecnologías, ya que está basada en una infraestructura tecnológica dinámica caracterizada por un alto grado de automatización, una rápida movilización de los recursos, una elevada capacidad de adaptación para atender a una demanda variable, así como virtualización avanzada. Entre estas ventajas destacan la flexibilidad, la escalabilidad y el ahorro de costes.

No se debe confundir computación en la nube con computación en grid (en red). La computación en grid es una forma de computación distribuida por la que "un súper computador virtual" está compuesto un cluster o conjunto de ellos que enlazan de ordenadores débilmente acoplados, actuando en concierto para realizar tareas muy grandes.

4.1.1. BENEFICIOS Y DESVENTAJAS

Algunos de estos beneficios son:

- Se integra con mucha mayor facilidad y rapidez con el resto de las aplicaciones.
- Implementación es más rápida y con menos riesgos, ya que se comienza a trabajar más rápido y no es necesaria una gran inversión.
- Como infraestructura presentan una mayor capacidad de adaptación, recuperación completa de pérdida de datos con copias de seguridad y reducción al mínimo de los tiempos de inactividad.
- Contribuye al uso eficiente de la energía. En las nubes, la energía consumida es sólo la necesaria, reduciendo notablemente el desperdicio.

Aunque presenta numerosas ventajas, repercute en ciertas desventajas:

- La disponibilidad de las aplicaciones está ligada a la disponibilidad de acceso a Internet.





- La confiabilidad de los servicios depende de la "salud" tecnológica y financiera de los proveedores de servicios en nube.
- La disponibilidad de servicios altamente especializados podría tardar meses o incluso años para que sean factibles de ser desplegados en la red.
- La madurez funcional de las aplicaciones hace que continuamente estén modificando sus interfaces, por lo cual la curva de aprendizaje en empresas de orientación no tecnológica tenga unas pendientes significativas, así como su consumo automático por aplicaciones.
- Seguridad. La información de la empresa debe recorrer diferentes nodos para llegar a su destino, cada uno de ellos y sus canales son un foco de inseguridad.
- Escalabilidad a largo plazo. A medida que más usuarios empiecen a compartir la infraestructura de la nube, la sobrecarga en los servidores de los proveedores aumentará, si la empresa no posee un esquema de crecimiento óptimo puede llevar a degradaciones en el servicio.

4.1.2. TIPOS DE SERVICIOS

El Cloud Computing ha permitido hacer uso de una infraestructura ya establecida de forma más fácil, eficiente, intuitiva y barata. La Infraestructura como Servicio (IaaS) es el eje central para que la "nube" pueda funcionar tal y como lo queremos. Es del IaaS donde se inicia la construcción de toda la infraestructura del Cloud Computing, y que da paso a la Plataforma como Servicio (PaaS) y al Software como Servicio (SaaS).

4.1.2.1. INFRAESTRUCTURA COMO SERVICIO (IAAS)

La infraestructura como Servicio (Infrastructure as a Service, IaaS, también llamado en algunos casos Hardware as a Service, HaaS) se encuentra en la capa inferior y es un medio de entregar almacenamiento básico y capacidades de cómputo como servicios estandarizados en la red.

Su principal beneficio es el ahorrando costos en mantenimiento y administración de equipos, permitiendo solo centrarse en la gestión de la información y de los clientes que tengan a su disposición.

El IaaS es perfecto para desarrolladores o compañías que se mantienen en constante evolución y crecimiento, ya que permite utilizar una larga lista de aplicaciones y servicios que cubrirán los requerimientos de hardware de la empresa evitando realizar excesivos costos y pagando sólo por los recursos que se utilicen.





El ejemplo comercial mejor conocido es Amazon Web Services, cuyos servicios EC2 que ofrece servicios de cómputo y S3 ofrece servicios de almacenamiento esenciales.

Otro ejemplo es Joyent cuyo producto principal es una línea de servidores virtualizados, que proveen una infraestructura en demanda altamente escalable para manejar sitios Web, incluyendo aplicaciones Web complejas escritas en Ruby o Rails, PHP, Python y Java.

4.1.2.2. PLATAFORMA COMO SERVICIO (PAAS)

La capa del medio, que es la Plataforma como Servicio (Platform as a Service, PaaS), es la encapsulación de una abstracción de un ambiente de desarrollo y el empaquetamiento de una serie de módulos o complementos que proporcionan, normalmente, una funcionalidad.

Plataforma como servicio podría consistir en un entorno que contiene una serie de sistemas, componentes o APIs preconfiguradas y listas para integrarse sobre una tecnología concreta de desarrollo.

Los servicios PaaS permiten gran flexibilidad, pero puede ser restringida por las capacidades que están disponibles a través del proveedor.

Los ejemplos comerciales incluyen Google App Engine, que sirve aplicaciones de la infraestructura Google, Windows Azure de Microsoft, una plataforma en la nube que permite el desarrollo y ejecución de aplicaciones codificadas en varios lenguajes y tecnologías como .NET, Java y PHP.

4.1.2.3. SOFTWARE COMO SERVICIO (SAAS)

El Software como Servicio (Software as a Service, SaaS) se encuentra en la capa más alta y caracteriza una aplicación completa ofrecida como un servicio, en demanda, donde solo una instancia del software corre en la infraestructura del proveedor y sirve a múltiples organizaciones de clientes.

Los ejemplos comerciales incluyen Google Apps que ofrecen servicios básicos de negocio como el e-mail. Otro ejemplo es la plataforma MS Office como servicio SaaS con su denominación de Microsoft Office 365, que incluye versiones online de la mayoría de las aplicaciones de esta suite ofimática de Microsoft.

4.1.3. TIPOS DE INFRAESTRUCTURAS

Existen tres tipos de modelos atendiendo a las necesidades y al modo de servicio ofrecido. Dependiendo del tipo de cliente que pueda usarlas estas son públicas, privadas o híbridas.





4.1.3.1. PRIVADAS

Las nubes privadas están en una infraestructura local en demanda manejada por un solo cliente que controla qué aplicaciones debe correr y dónde. Son propietarios del servidor, red, y disco y pueden decidir qué usuarios están autorizados a utilizar la infraestructura. Las nubes privadas, las plataformas se encuentran dentro de las instalaciones del usuario de la misma y no suele ofrecer servicios a terceros.

Como ventaja de este tipo de nubes, al contrario que las públicas, es la localización de los datos dentro de la propia empresa, lo que conlleva a una mayor seguridad de estos, corriendo a cargo del sistema de información que se utilice. Incluso será más fácil integrar estos servicios con otros sistemas propios.

Sin embargo, como inconveniente se encuentra la inversión inicial en infraestructura física, sistemas de virtualización, ancho de banda y seguridad, lo que llevará a su vez a pérdida de escalabilidad y desestabilidad de las plataformas, sin olvidar el gasto de mantenimiento que requiere. Esta alta inversión supondrá un retorno más lento de la inversión.

4.1.3.2. PÚBLICAS

Se manejan en servidores externos al usuario, y los trabajos de muchos clientes diferentes pueden estar mezclados en los servidores, los sistemas de almacenamiento y otras infraestructuras de la nube. Los usuarios finales no conocen qué trabajos de otros clientes pueden estar corriendo en el mismo servidor, red, discos como los suyos propios

La ventaja más clara de las nubes públicas es la capacidad de procesamiento y almacenamiento sin instalar máquinas localmente, por lo que no tiene una inversión inicial o gasto de mantenimiento en este sentido, si no que se paga por el uso. La carga operacional y la seguridad de los datos recae íntegramente sobre el proveedor del hardware y software, debido a ello, el riesgo por la adopción de una nueva tecnología es bastante bajo. El retorno de la inversión se hace rápido y más predecible con este tipo de nubes. A veces puede resultar difícil integrar estos servicios con otros sistemas propios.

4.1.3.3. HÍBRIDAS

Las nubes híbridas combinan los modelos de nubes públicas y privadas. Esto permite a una empresa mantener el control de sus principales aplicaciones, al tiempo de aprovechar el Cloud Computing en los lugares donde tenga sentido.

Una nube híbrida tiene la ventaja de una inversión inicial más moderada y a la vez contar con SaaS, PaaS o IaaS bajo demanda. En el momento necesario, utilizando las APIs de las distintas plataformas públicas existentes, se tiene la





posibilidad de escalar la plataforma todo lo que se quiera sin invertir en infraestructura.

4.2. VIRTUALIZACIÓN

4.2.1. HIPERVISORES

Un hipervisor es una plataforma que permite aplicar diversas técnicas de control de virtualización para utilizar, al mismo tiempo, diferentes sistemas operativos. Son aplicaciones que presentan a los sistemas operativos virtualizados (sistemas invitados) una plataforma operativa virtual (hardware virtual), a la vez que ocultan a dicho sistema operativo virtualizado las características físicas reales del equipo sobre el que operan. Son también los encargados de monitorizar la ejecución de los sistemas operativos invitados.

Existen tres tipos de hipervisores en el mercado:

- Hipervisor de tipo 1
- Hipervisor de tipo 2.
- Hipervisor híbrido.

Un hipervisor (independientemente de su tipo) no es más que una aplicación en niveles que abstrae el hardware de la máquina de sus huéspedes. De esta manera, cada huésped ve una VM en lugar del hardware real. Ahora analicemos de manera genérica la composición interna de un hipervisor y también la forma en que se presenta ante las VM (o sistemas operativos huésped). Requiere de una cantidad reducida de elementos para hacer arrancar un sistema operativo huésped: una imagen del núcleo para arrancar, una configuración (como, por ejemplo, direcciones IP y cantidad de memoria a usar), un disco y un dispositivo de red. El disco y el dispositivo de red suelen mapearse hacia el disco físico y el dispositivo de red de la máquina. Finalmente, se requiere de un conjunto de herramientas del huésped para iniciar un huésped y luego gestionarlo.

4.2.1.1. HIPERVISOR TIPO I

También denominado nativo, unhosted o bare metal (sobre el metal desnudo), el hipervisor se ejecuta directamente sobre el hardware físico. Es el hipervisor quien controla todos los accesos directos a hardware.

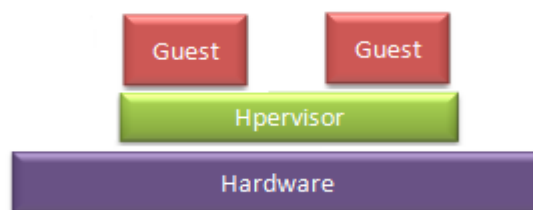


Ilustración 2 Hipervisor Tipo I





Algunos de los hipervisores tipo I más conocidos son los siguientes: VMware ESXi (gratis), VMware ESX (de pago), Xen (libre), Citrix XenServer (gratis), Microsoft Hyper-V Server (gratis).

Los hipervisores de tipo I a su vez pueden ser de dos tipos:

- **Monolíticos:** emulan hardware para sus máquinas virtuales. Ante una llamada de un sistema virtualizado el hardware emulado debe interceptar la llamada y redirige las llamadas hacia los drives del dispositivo que opera en el hipervisor, estos a su vez enrutan la llamada hacia el dispositivo físico.
- **De MicroKernel:** el hipervisor se reduce a una capa de software muy sencilla, cuya única funcionalidad es la de particionar el sistema físico entre los diversos sistemas virtualizados. No requiere drivers específicos para acceder al hardware. Este tipo aumenta el rendimiento al reducir código intermedio y el número de cambios de contexto necesarios, y aumenta la estabilidad de los sistemas al haber menos componentes, y seguridad al reducir la superficie de ataque del hipervisor.

4.2.1.2. HIPERVISOR TIPO II

También denominado hosted, el hipervisor se ejecuta sobre un sistema operativo. Las máquinas virtuales se ejecutan en un tercer nivel por encima del hipervisor.

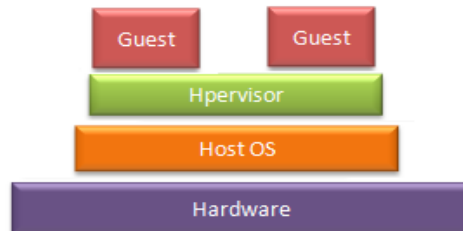


Ilustración 3 Hipervisor Tipo II

Se utilizan mucho para virtualización orientada a la ejecución multiplataforma de software. Algunos de los hipervisores tipo 2 más utilizados son los siguientes: Oracle: VirtualBox (gratis), VirtualBox OSE (libre), VMware: Workstation (de pago), Server (gratis), Player (gratis), QEMU (libre), Microsoft: Virtual PC, Virtual Server.

4.2.1.3. HIPERVISOR HÍBRIDO

En este modelo tanto el sistema operativo anfitrión como el hipervisor interactúan directamente con el hardware físico.

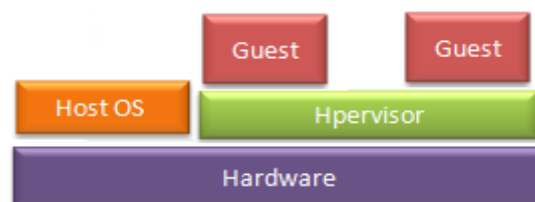


Ilustración 4 Hipervisor Tipo Híbrido





4.3. VALORACIÓN DE APLICACIONES EXISTENTES

Con el objetivo de implementar una solución siguiendo el paradigma de Cloud-Computing, hemos basado nuestra aproximación en aplicaciones ya existentes que cuentan con una buena aceptación en la comunidad de usuarios y han demostrado buenos resultados. Se han valorado los casos particulares de Boinc y Wolfram — Alpha.

La Infraestructura Abierta de Berkeley para la Computación en Red (BOINC) es una infraestructura para la computación distribuida, diseñada para ser una plataforma libre. Está dirigido a proyectos que requieren una gran capacidad de cálculo a través de la utilización de ordenadores personales distribuidos alrededor del mundo, realizando los cálculos en los ciclos de CPU o GPU que no son utilizados. BOINC consiste en un servidor y un cliente que se comunican para distribuir y procesar el trabajo. Cada ordenador es asignado a un único proyecto⁵. Wolfram—Alpha tiene la apariencia de motor de búsqueda pero no busca respuestas a las preguntas en un conglomerado de páginas web o documentos, sino que realiza cálculos en su servidor y envía la respuesta para que sea visualizada por el usuario a través de una interfaz web.

4.3.1. CONCLUSIONES DEL ESTUDIO

Se concluyó que no existía una solución completa que proporcionara los servicios que se querían ofrecer pero si existían soluciones que los cubrían parcialmente, por lo que se realizó un análisis de las herramientas existentes como componentes para implementar una solución completa.

4.4. NUEVAS ESTRATEGIAS Y SOLUCIONES

Como parte del diseño arquitectónico de la aplicación surge la necesidad de utilizar: un hipervisor, un gestor de colas PBS para la capa de integración y una herramienta de cálculo potente con interfaz web.

4.4.1. HIPERVISORES: VMWARE Y VIRTUALBOX.

La versión analizada de VirtualBox es de código abierto y sujeto a la licencia GPL. Además, es funcionalmente idéntico en todas las plataformas de acogida y utiliza el mismo archivo y formato de imagen. Esto permite ejecutar la misma imagen de máquina virtual en instancias de VirtualBox que se ejecutan sobre diferentes sistemas operativos por lo que es fácil exportar las imágenes entre entornos, ofreciendo mayor portabilidad. VirtualBox está en constante desarrollo y actualización por lo que nuevas funcionalidades son implementadas





regularmente. Además, cuenta con una API que nos permite integrarlo fácilmente a nuestra solución.

VMware también cuenta con una versión gratis con funcionalidades similares a VirtualBox, pero al realizar una comparativa de las tecnologías se descarta dado que la filosofía de la solución es utilizar software de código abierto.

4.4.2. GESTORES PBS

PBS (*Portable Batch System*) es un sistema flexible de balanceo de carga y planificación de tareas, inicialmente fue desarrollado para administrar recursos computacionales de la NASA. PBS ha sido el líder en la administración de recursos y considerado el estándar de facto para los sistemas de planificación bajo sistemas Linux.

Actualmente existen dos versiones de PBS: una versión antigua de código abierto Altair OpenPBS y la versión comercial Altair PBS Pro. Actualmente OpenPBS y PBS Pro se incluyen en algunos *toolkits* para la instalación automática de *cluster*. PBS Pro provee algunas funcionalidades y beneficios para los administradores del *cluster*.

El gestor PBS Torque (Tera-scale Open-source Resource and Queue manager) es un administrador de recursos que nos provee de mecanismos para controlar trabajos sobre distintos nodos de un cluster. Permite gestionar el estado de los nodos y su utilización. Basado en OpenPBS, es el sistema de colas batch de código abierto líder en el mercado. Soporta el procesamiento en paralelo de múltiples batches y colas de trabajo interactivas.

4.4.3. HERRAMIENTAS DE CÁLCULO

Otro de los objetivos de la solución es ofrecer una interfaz web. El estudio de herramientas disponibles se enfocó en encontrar un entorno interactivo que se pudiera integrar a la infraestructura existente y que contara con soporte multiusuario, pero que a la vez fuera una poderosa herramienta de cálculo. Considerando las restricciones mencionadas se evaluaron las alternativas de Sage e iPython.

Sage fue diseñado desde un principio como solución escalable a grandes servidores multiusuario centralizados (ejemplo sagenb.org con cerca de 76000 cuentas), pero con la desventaja de que el mismo binario que se ejecuta en grandes servidores es utilizado para ejecuciones que no requieren tantos recursos, lo que se refleja en un bajo rendimiento en ciertos entornos. Sage tiene licencia GPL.





IPython es multiplataforma, software libre, tiene una enorme comunidad detrás, un desarrollo constante y bien organizado, permite paralelismo y es extremadamente potente. Nació de las manos de investigadores y en un ámbito científico, por lo que su comunidad es mayormente académica al igual que nuestros usuarios. IPython trabaja con un sistema de archivos estándar accesible trivialmente desde el sistema operativo, facilitando la realización de operaciones rutinarias en el análisis de datos y simulación numérica como acceder a scripts locales o ver archivos arbitrarios. IPython incluye una serie de opciones para dar formato a los documentos: capacidades multihoja, preámbulo de estilo Látex, metadatos por celda, navegación outline-level, y apoyo validación / reproducibilidad entre otras. IPython tiene licencia BSD. Además permite añadir nuevas extensiones como Matlab, SQL y Physicsy. En lo referente a paralelismo, iPython lo abstrae de una manera muy general, lo que permite trabajar sobre muchos estilos diferentes de paralelismo, incluyendo: Programa individual (SPMD) paralelismo de datos múltiples, Programa múltiple, (MPMD) paralelismo de datos múltiples, Message Passing usando MPI, combinaciones de estos enfoques y un usuario personalizado definido enfoques. Lo más importante, IPython permite todo tipo de aplicaciones paralelas para ser desarrolladas, ejecutadas, depuradas y supervisadas interactivamente. Por lo tanto, la I en IPython, hace referencia a lo interactivo.

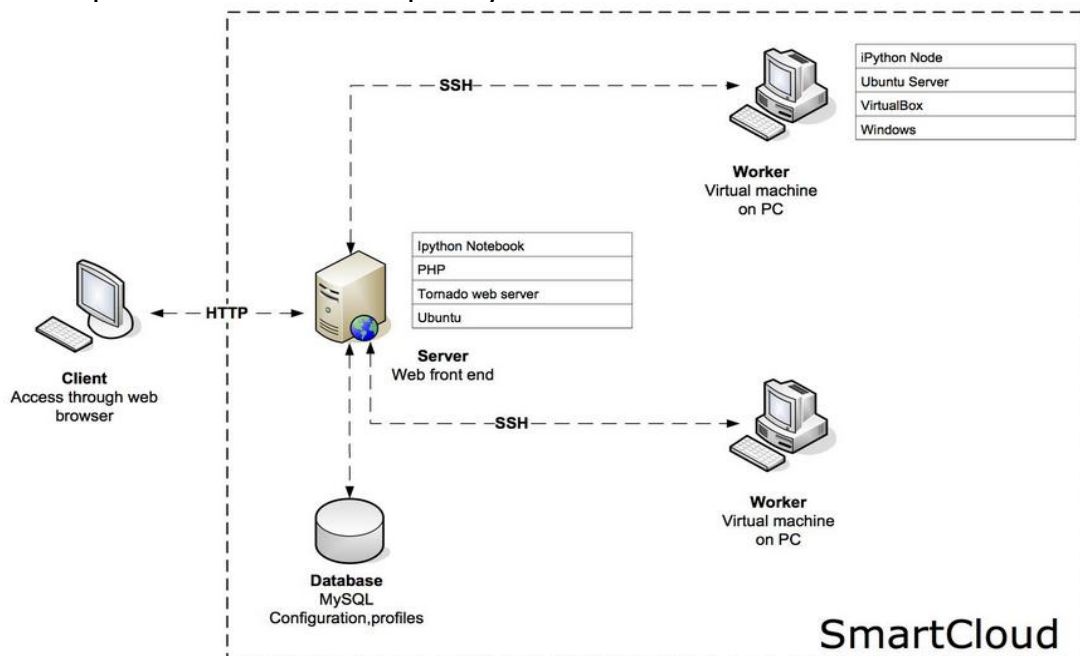




5. ARQUITECTURA DEL SISTEMA

5.1. APROXIMACIÓN DE ALTO NIVEL

El siguiente diagrama muestra una aproximación de alto nivel de los elementos que componen la solución completa y como son sus interacciones.



A continuación se detallará la arquitectura de la solución.

5.2. ARQUITECTURA POR CAPAS

Después de un análisis en profundidad llegó a la conclusión de implementar una arquitectura por capas por su beneficio modular.

- Capa Inferior: formada por los nodos de procesamiento conectados a una red para poder ser accesibles y gestionables por el servidor.
- Capa Intermedia de integración: comunica la capa superior con la inferior. Gestiona la distribución de los trabajos generados desde la capa superior.





SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

- Capa Superior: contiene el frontend para la interacción con el usuario. En esta capa se encuentra la aplicación de cálculo que se despliega. Cuando la herramienta necesite realizar funciones sobre el cluster realizará llamadas a capas inferiores, lo que se refleja en un aumento en la carga del procesamiento del cluster.

Dada a la diversidad de alternativas para implementar estas capas, con el previo análisis de cada una, se llegó a las conclusiones de diseño:

- En la capa inferior para definir los nodos de procesamiento se decidió realizarlo mediante virtualización para homogeneizar las máquinas a usar. Como ya se ha comentado anteriormente, en cada nodo se instala una máquina virtual y sobre está una misma imagen. Se decidió utilizar Oracle VM VirtualBox, un hipervisor de tipo 2, dada su facilidad de instalación y configuración y sus garantías de acceso al hardware físico a través del sistema operativo host. Para el sistema operativo a virtualizar se decidió Ubuntu (Linux) por ser software libre.

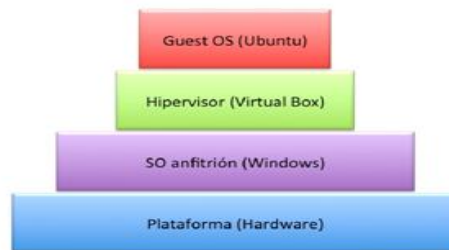


Ilustración 5 Arquitectura de capas - Capa inferior

- Para la capa intermedia, se decidió utilizar un gestor de trabajos PBS Torque que nos permite una gestión de procesamiento transparente. Además es el sistema de colas batch de código abierto líder en el mercado. Esta herramienta es un fork de OpenPBS y es gratuita.
- Para la capa superior, se eligió iPython gracias a su entorno completo para la computación interactiva y exploratoria. Además, presenta una arquitectura robusta construida alrededor del concepto de intérprete de Python pero a la vez sencilla y eficaz [2]. También se tuvo en cuenta el interés creciente en esta herramienta [17].

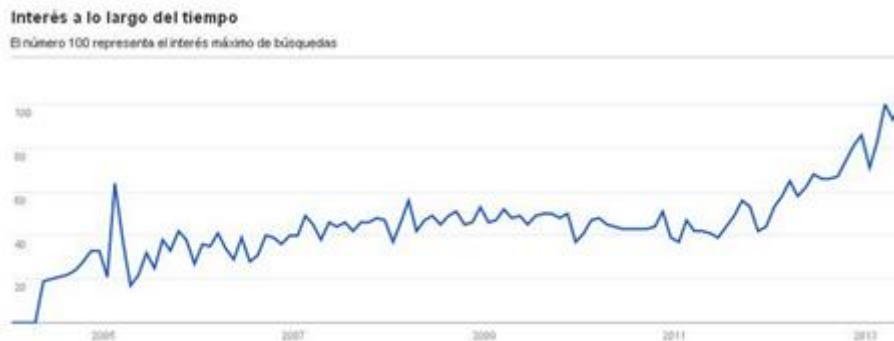


Ilustración 6 - Evolución del interés en iPython





5.3. IMPLEMENTACIÓN DE LAS CAPAS

La siguiente figura ilustra la arquitectura general de la aplicación y la comunicación entre las capas.

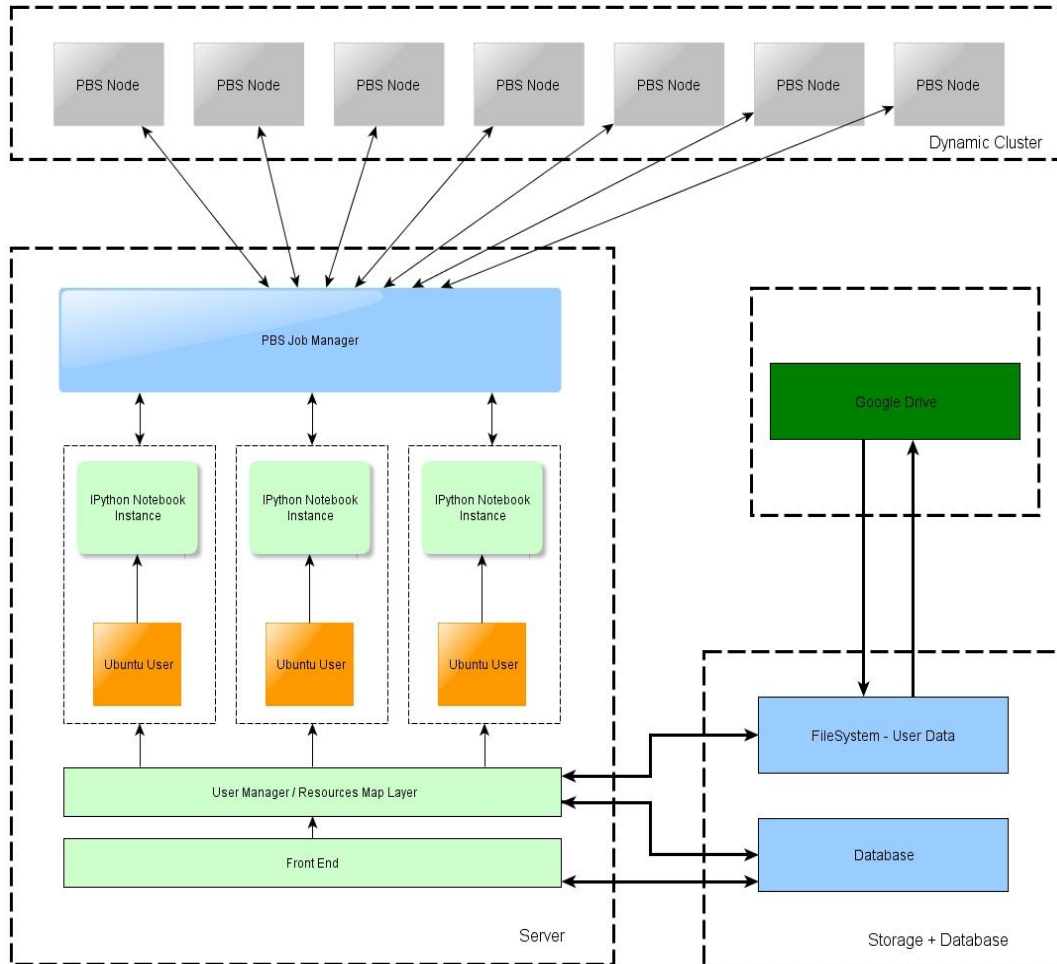


Ilustración 6 Arquitectura de la solución

5.3.1. CAPA INFERIOR: NODO DE PROCESAMIENTO

Como mencionamos antes, la idea es levantar nodos de procesamiento cuando el ordenador no esté siendo utilizado, y aprovechar ese tiempo para ejecutar las funciones iPython.

Para el nodo se cuenta con una única imagen .iso de Ubuntu 10.04 con las librerías para ejecutar iPython y establecer la comunicación con el cluster y la base de datos.

Partiendo de la especificación de diseño y la necesidad de administrar el momento en el cual los nodos son activados, se necesitó añadir tareas (controlador de tareas) que levante la máquina virtual cuando detecte que no se





está utilizando, la baje cuando reconozca un inicio de sesión válido, informando de su estado al gestor de tareas Torque y se guarde la información en la base de datos.

El esquema general del nodo sería el siguiente:



Ilustración 7 Esquema del Nodo

5.3.2. CAPA INTERMEDIA: PBS TORQUE Y NFS

5.3.2.1. GESTOR PBS TORQUE

Como capa intermedia se utiliza el gestor de trabajos Torque que es una implementación de un PBS (Potable batch system). Se ha definido una única cola en esta herramienta en la que convergen todas las peticiones de las instancias de iPython [12].

5.3.2.1.1. CARACTERÍSTICAS MÁS IMPORTANTES DEL SISTEMA PBS

- Múltiples interfaces de usuario: Proveen una interfaz gráfica de usuario para realizar peticiones por lotes y ejecución de tareas.
- Listas de seguridad y control de acceso: El administrador puede permitir o denegar el acceso al sistema PBS basándose en el nombre de usuario, grupo, nodo o dominio de red.
- Registro de tareas: *logs* detallados de las actividades del sistema mediante el análisis de utilización por usuario, por grupo y por nodo.
- Soporte de tareas paralelas: Permite la utilización de librerías de programación paralela como MPI, PVM, y HPF. Se puede planificar la ejecución de aplicaciones sobre un computador de un sólo procesador o mediante la utilización de múltiples computadores.
- Monitoreo del sistema: Mediante una interfaz gráfica permite realizar un monitoreo completo del ambiente distribuido.
- Soporte para *grids*: Provee tecnología para *grids* computacionales y la integración del *toolkit* Globus.





- API de desarrollo: Permite desarrollar aplicaciones que integran PBS como una de sus herramientas para requerimientos de balanceo de carga.
- Nivel de carga automático: Provee diversas formas de distribuir la carga en los computadores que conforman el *cluster*, basados en la configuración de hardware, disponibilidad de recursos, actividad del teclado y el manejo de políticas locales.
- *Distributed clustering*: Permite la disponibilidad de recursos de *cluster* y otros sistemas distribuidos en una red WAN.
- Ambiente común de usuario: Ofrece al usuario una visión común de las tareas entregadas y solicitadas, el estado del sistema y seguimiento de tareas.
- Prioridad de tareas: Permite a los usuarios especificar prioridades para la asignación de recursos y ejecución de sus tareas.
- Disponibilidad para diferentes plataformas: Permite el soporte de Windows 2000 y XP, junto con la mayoría de versiones de UNIX y Linux, desde estaciones de trabajo y servidores hasta supercomputadores.

5.3.2.1.2. IMPORTANCIA DE UN GESTOR PBS

Al utilizar un gestor de jobs, se simplifica la integración de ambas capas dado que es una interfaz transparente para la capa superior. En otras palabras, iPython solamente ve un único punto donde realizar peticiones/envíos de trabajos, pero es Torque quien se encarga de distribuir los trabajos entre los recursos disponibles según políticas de optimización.

5.3.2.2. NETWORK FILE SYSTEM (NFS)

Para el correcto funcionamiento del clúster, se ha implementado un NFS (network file system) que funciona como directorio de scratch para almacenar archivos temporales, trabajos, y salida de trabajos ya terminados.

En una arquitectura de cluster, es necesario disponer de al menos un directorio de trabajo que funcione de "scratch" para que los diferentes nodos puedan compartir información. Dicho directorio debe poder ser accesible por todos los nodos con permiso de lectura y escritura.

En esta solución, hemos configurado un servidor Network File System(NFS) para satisfacer esta necesidad.

El NFS, es un protocolo de nivel de aplicación, según el Modelo OSI. Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local. Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales.





5.3.2.2.1. CARACTERÍSTICAS

El sistema NFS está dividido al menos en dos partes principales: un servidor y uno o más clientes. Los clientes acceden de forma remota a los datos que se encuentran almacenados en el servidor.

Las estaciones de trabajo locales utilizan menos espacio de disco debido a que los datos se encuentran centralizados en un único lugar pero pueden ser accedidos y modificados por varios usuarios, de tal forma que no es necesario replicar la información.

Los usuarios no necesitan disponer de un directorio "home" en cada una de las máquinas de la organización. Los directorios "home" pueden crearse en el servidor de NFS para posteriormente poder acceder a ellos desde cualquier máquina a través de la infraestructura de red.

También se pueden compartir a través de la red dispositivos de almacenamiento como disqueteras, CD-ROM y unidades ZIP. Esto puede reducir la inversión en dichos dispositivos y mejorar el aprovechamiento del hardware existente en la organización.

5.3.2.3. DETALLES DE IMPLANTACIÓN DE PBS EN SMARTCLOUD

La capa intermedia de SmartCloud está compuesta por los dos componentes antes mencionados.

Se definió un NFS que comparte la carpeta `\home\server\shared` entre nodos y servidor (la ruta es la misma en ambos casos).

Se configuró iPython y sus correspondientes componentes iPcontroler e iPengine para que el directorio de trabajo sea `\home\server\shared`. Como resultado de esto, cuando iPython necesita realizar funciones con el clúster, utiliza ésta carpeta como scratch.

Todas las instancias de iPython que se encuentran paralelamente en ejecución utilizan la misma cola de trabajo definida en el gestor PBS (cola 'batch') para el envío de trabajo. El gestor PBS es el encargado de distribuir los trabajos entre los nodos. El funcionamiento es el siguiente:



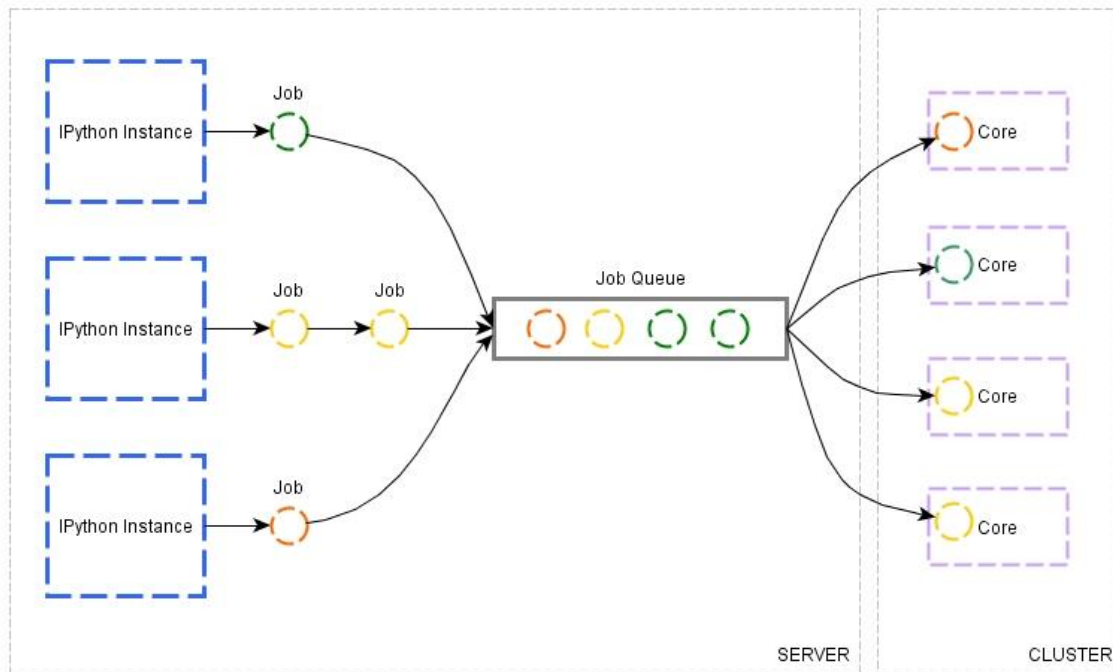


Ilustración 8 Funcionamiento del gestor PBS

5.3.3. CAPA SUPERIOR: FRONTEND Y HERRAMIENTA DE CÁLCULO IPYTHON

Para la capa superior se define un frontend con gestión de usuarios para el acceso y uso de la herramienta de cálculo iPython.

5.3.3.1. FRONTEND

Para facilitar esta gestión y no guardar información de la cuenta de los usuarios, se decidió utilizar un proveedor de OpenID.

La autenticación federada es un mecanismo de seguridad descentralizado que no está ligado a ningún proveedor determinado y OpenID un estándar abierto que implementa la autenticación federada.

En el terreno de OpenID, existen dos tipos de usuarios:

- Proveedor de identidad o servidor: servicio que aloja identidades de OpenID y puede autenticar a los usuarios con credenciales creadas allí.
- Parte de confianza o consumidor: cualquier aplicación que ha delegado su autenticación a un proveedor de OpenID (el caso de nuestro frontend).

Un servicio de acceso federado hace realidad un inicio de sesión único universal, ofrece más opciones y una experiencia del usuario más flexible y adecuada.





Considerando los dos proveedores fundamentales Google y Yahoo, se analizaron las estadísticas para saber cuál era el más utilizado [23], y se decidió utilizar la OpenID que ofrece Google.

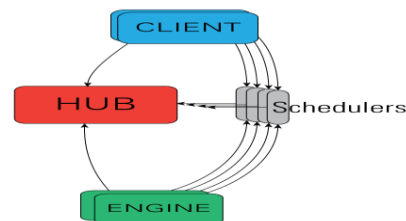
Una vez que se realiza la autenticación exitosa de un usuario por parte de Google, se analiza si el usuario autenticado tiene permisos para ejecutar el notebook de iPython dentro de nuestro frontend.

5.3.3.2. IPYTHON

La herramienta de cálculo IPython proporciona un kit de herramientas enriquecidas para ayudar a utilizar Python al máximo interactivamente. Cuenta con un poderoso intérprete shell de Python (basado en terminal y Qt), compatibilidad para visualización de datos interactiva y el uso de kits de herramientas de GUI, intérpretes flexibles para cargar proyectos y herramientas para la computación de alto nivel e interactiva paralela y un notebook basado en la web con las mismas características básicas pero con compatibilidad para media enriquecida, texto, código, expresiones matemáticas y trazados en línea. En este proyecto se integra el notebook de iPython con el frontend.

La arquitectura IPython consta de cuatro componentes:

- IPython engine.
- IPython hub.
- IPython schedulers.
- Controller client.



Estos componentes están en el paquete IPython.parallel y se instalan con IPython.

- *IPython engine*, es una instancia de Python que toma comandos Python en una conexión de red. Con el tiempo, el motor IPython será un intérprete IPython completo, pero por ahora, es un intérprete de Python regular. El motor también puede manejar los objetos de Python entrantes y salientes enviados a través de una conexión de red. Cuando se ponen en marcha varios motores, la computación paralela y distribuida hace posible. Una característica importante de un motor IPython es que bloquea mientras se ejecuta el código de usuario. Leer para cómo el controlador IPython resuelve este problema es exponer una API asíncrona limpia para el usuario.
- *IPython hub*, Este es el proceso que realiza un seguimiento de las conexiones del motor, los schedulers, los clientes, así como todas las solicitudes y resultados de las tareas. La función principal del Hub es





facilitar las consultas del estado del clúster, y reducir al mínimo la información necesaria para establecer las múltiples conexiones que intervienen en la conexión de nuevos clientes y motores.

- *IPython schedulers*, todas las acciones se pueden realizar sobre la marcha del motor a través de un *schedulers*.
- *Controller*, un conjunto de procesos para que los motores IPython y los clientes puedan conectarse. El controlador se compone de un *hub* y una colección de *schedulers*. Esto se ejecuta normalmente en procesos separados pero en la misma máquina que el *hub*, pero se pueden ejecutar en cualquier lugar de temas locales o en las máquinas remotas.

Python es un lenguaje interpretado y viene con su propio intérprete. Python se puede utilizar de modo interactivo en el terminal de Python, y esto es considerado como uno de los puntos fuertes de Python ya que fomenta la "computación exploratoria" que permite a los programadores probar sencillos pasos y algoritmos antes de intentar escribir funciones y módulos.

IPython es una shell mejorada para Python que es rico en características y hace la vida más fácil para aquellos que utilizan Python para la computación científica. Partida IPython con la opción-pylab incluye NumPy y SciPy en el espacio de nombres estándar de modo que no es necesario importar estos módulos antes de utilizar las funciones de estos módulos (Esto es conveniente para el uso interactivo, siempre incluyen estos módulos en los scripts, ya que se ser interpretado y ejecutado por Python y no por IPython).

Además de importar automáticamente NumPy y SciPy, IPython ofrece otras características que son útiles cuando se utilizan Python interactiva:

- Ayuda en línea es muy poderosa. Finalización automática pulsando la tecla Tab (similar a la función de shell bash en Linux), poniendo fin a un comando con el signo de interrogación y pulsar enter muestra ayuda sobre el comando específico
- Funciones mágicas que dan le permiten hacer cosas como ejecutar un script de Python
- Acceso al sistema de archivos

5.3.3.2.1. NOTEBOOK

El notebook de IPython es una interfaz web para IPython, inspirada en los notebooks de Mathematica y Sage. En la parte de la interfaz utiliza MathJax para las fórmulas matemáticas y el editor CodeMirror.

El notebook proporciona un entorno de trabajo para la computación interactiva que combina la ejecución de código con la creación de un documento computacional en vivo. Estos archivos portátiles pueden contener texto arbitrario, fórmulas matemáticas, el código de entrada, los resultados, los gráficos, videos y





cualquier otro tipo de medios de comunicación que un navegador web moderno es capaz de mostrar.

El Notebook IPython consta de dos componentes relacionados:

- Un formato de documento portátil basado en JSON para grabar y distribuir código Python y texto enriquecido.
- Una interfaz de usuario basada en web para la creación de documentos y portátiles que ejecutan.

Estas son algunas de las características del notebook de IPython:

- Lista de notebooks.
- Edición y movimiento de celdas, inserción y eliminación arbitraria.
- Gráficos y figuras integrados con el modo --pylab inline
- Autocompletado de código.
- Celdas con encabezados, puro HTML o Markdown.
- Posibilidad de utilizar R, Octave, Cython.
- Enlaces a la documentación del notebook, IPython, Python, NumPy, Scipy, SymPy y matplotlib.
- Ayuda integrada.
- Exportación a archivo .ipynb, basado en JSON, para compartir notebooks.
- Atajos de teclado

El notebook de iPython es la combinación de:

- Una sesión interactiva conectada a un kernel IPython, controlado por una aplicación web que puede enviar información a la consola y mostrar muchos tipos de salida (texto, gráficos, matemáticas y más). Este es el mismo núcleo utilizado por la consola Qt, pero en este caso la consola de web envía de entrada en las células persistentes que se puede editar en el lugar en lugar de la terminal de estilo de desplazamiento vertical utilizado por la consola Qt.
- Un documento que puede salvar a las entradas y salidas de la sesión, así como el texto adicional que acompaña al código, pero no es para la ejecución. De esta manera, los archivos portátiles sirven como un registro computacional completa de una sesión incluyendo texto explicativo y matemáticas, código y cifras resultantes. Estos documentos son internos archivos JSON y se guardan con la extensión ipynb.

5.3.3.3. DETALLES DE IMPLANTACIÓN DE NOTEBOOK EN SMARTCLOUD

En principio iPython no estaba implementado para poder ser utilizado por varios usuarios de manera simultánea bajo un mismo servidor, razón por la cual tuvimos que modificarlo para añadir esta funcionalidad.





Una de las principales características del notebook es que se puede configurar su funcionamiento mediante la creación de un perfil. Al crear este perfil se genera un fichero `ipython_notebook.py` mediante el cual se puede redefinir las características del notebook. Por ejemplo en este fichero se indica la carpeta donde queremos guardar todo el trabajo generado, así como el puerto de donde queremos que se ejecute entre otras. Al poder definir el puerto, es lo que nos permite poder lanzar varias instancias del notebook, ya que lo hace desde puertos diferentes.

Un perfil es un directorio que contiene los archivos de configuración y la información para la conexión con las aplicaciones paralelas.

La idea es que los usuarios mantengan un conjunto de archivos de configuración para diferentes propósitos: uno para hacer cálculos numéricos con NumPy y SciPy y otra para hacer la computación simbólica con SymPy.

Los perfiles hacen más fácil mantener archivos de configuración separados, registros e historias de cada uno de estos propósitos.

Al hacer esto para un solo usuario fue fácil generalizar este procedimiento para múltiples usuarios, teniendo en cuenta que cada uno tendría su propia instancia de notebook ejecutándose sobre un puerto diferente.

El Notebook IPython cubre una amplia gama de casos de uso incluyendo la exploración simple y aprender Python, análisis de datos y visualización, simulación y computación paralela. Los archivos resultantes contienen un registro completo de los cálculos que se realizan y se pueden compartir con otros usuarios IPython.

El Notebook IPython se puede utilizar como una aplicación local, pero es ideal para paralelizar trabajos en diferentes nodos.





6. INSTALACIÓN Y CONFIGURACIÓN DE LOS COMPONENTES INTERNOS DE LA SOLUCIÓN

6.1. GESTOR PBS TORQUE

Compilar la distribución en la maquina que va a funcionar como TORQUE server, la cual va a monitorizar y controlar todos los nodos de cómputo ejecutando el `pbs_server` daemon[18].

- El paquete compilado funciona únicamente en nodos de cómputo con arquitectura similar. Para nodos con arquitecturas diferentes se debe compilar el paquete independientemente

1. Descargar el archive de TORQUE de:

<http://clusterresources.com/downloads/torque>.

El código fuente puede ser también descargado utilizando Subversión de <svn://clusterresources.com/torque/>.

2. Extraer el archivo empaquetado y navegar al directorio:

```
>tar -xzvf torque-2.3.4.tar.gz  
>cd torque-2.3.4/
```

3. Configurar el paquete:

```
> ./configure
```

4. Ejecutar `make` y `make install`:

```
>make  
> sudo make install
```





Por defecto, `make install` crea un directorio en `/var/spool/torque`. Este directorio se denomina `$TORQUEHOME`. `$TORQUEHOME` tiene varios sub-directorios, incluyendo:

`server_priv/`, `server_logs/`, `mom_priv/`, `mom_logs/`

Y otros directorios utilizados en la configuración y ejecución de TORQUE.

TORQUE 2.0p2 y posterior incluye un archivo `torque.spec` para compilar otra RPMs personalizada.

6.1.1. NODOS DE CÓMPUTO

Utilizar el "Cluster Resources tpackage system" para crear archivos auto-extraíbles que pueden ser distribuidos e instalados en nodos de cómputo. Los tpackages son personalizables. Consultar el archivo `INSTALL` para opciones y funcionalidades adicionales.

6.1.2. PARA CREAR TPACKAGES

1. Ejecutar `make packages`:

```
>make packages
Building ./torque-package-clients-linux-i686.sh ...
Building ./torque-package-mom-linux-i686.sh ...
Building ./torque-package-server-linux-i686.sh ...
Building ./torque-package-gui-linux-i686.sh ...
Building ./torque-package-devel-linux-i686.sh ...
Done.
```

```
The package files are self-extracting packages that
can be copied
and executed on your production machines.  Use --
help for options.
```

2. Copiar los paquetes a una carpeta compartida.

```
>cp torque-package-mom-linux-i686.sh
/shared/storage/
>cp torque
```

3. Instalar los tpackages en los nodos de cómputo

Se recomienda utilizar una Shell remota, como SHH, para instalar los tpackages en sistemas remotos. Configurar claves SSH compartidas por comodidad.





Cluster Resources recommends that you use a remote shell, such as SSH, to install tpackages on remote systems. Set up shared SSH keys if you do not want to supply a password for each host.

- El único paquete requerido para los nodos de cómputo es mom-linux. Los paquetes adicionales son recomendados para poder utilizar comandos en el cliente para enviar trabajos desde los nodos de cómputo.

A continuación se muestra un ejemplo de cómo copiar e instalar mom-linux de manera distribuida:

```
>for i in node01 node02 node03 node04 ; do scp
torque-package-mom-linux-i686.sh ${i}:/tmp/. ; done
> for i in node01 node02 node03 node04 ; do scp
torque-package-clients-linux-i686.sh  ${i}:/tmp/. ;
done
>for i in node01 node02 node03 node04 ; do ssh ${i}
/tmp/torque-package-mom-linux-i686.sh  --install  ;
done
>for i in node01 node02 node03 node04 ; do ssh ${i}
/tmp/torque-package-clients-linux-i686.sh  --install
; done
```

4. Copiar tpackage a los nodos.

```
>prcp torque-package-linux-i686.sh
noderange:/destinationdirectory/
```

5. Instalar el tpackage.

```
pshnoderange/tmp/torque
```

Es posible utilizar TORQUE server como nodo de cómputo e instalar pbs_mom junto con el demonio pbs_server.

6.1.3. ESTABLECER TORQUE COMO SERVICIO (OPCIONAL)

Los scripts de inicio se encuentran en la carpeta contrib/init.d/ del paquete source.

Ejecutar comoroot

```
cp contrib/init.d/debian.pbs_mom /etc/init.d/pbs_mom
>update-rc.d pbs_mom
```

Esta opción puede ser añadida al paquete auto-extraíble. Para más detalles consultar el archivo INSTALL.





6.1.4. CONFIGURACIÓN BÁSICA

Por defecto, makeinstall instala los archivos en:

```
/usr/local/bin, /usr/local/lib, /usr/local/sbin, /usr/local/in  
clude, y /usr/local/man .
```

Es posible especificar un prefijo de instalación diferente de /usr/local utilizando `--prefix` como un argumento a `./configure`, por ejemplo:

```
./configure --prefix=$HOME
```

Verificar que el sistema tiene las variables de entorno configuradas y que puede encontrar los binarios y librerías compartidas de TORQUE.

Para configurar la ruta de las librerías, añadir el directorio en donde se instalarán las librerías de TORQUE. Por ejemplo, si las mismas se instalarán en /opt/torque/lib ejecutar lo siguiente:

```
> set LD_LIBRARY_PATH=$(LD_LIBRARY_PATH):/opt/torque/lib  
> ldconfig
```

Se recomienda que el administrador de TORQUE sea root

6.1.5. INICIAR/CONFIGURAR TORQUE EN EL SERVIDOR (PBS_SERVER)

`$TORQUEHOME/server_priv/` contiene configuración y otra información necesaria para el `pbs_server`. Uno de los archivos en este directorio `serverdb`. `serverdb` contiene parámetros de configuración para `pbs_server` y sus colas. Para que se ejecute `pbs_server`, `serverdb` tiene que estar inicializado.

`serverdb` puede ser inicializado de dos maneras:

```
pbs_server -t create
```

Ejecutar `./torque.setup` desde el directorio de compilación.

Reiniciar `pbs_server` luego de inicializar `serverdb`.

```
> qterm  
> pbs_server  
pbs_server -t create
```

La opción '-t create' le indica a `pbs_server` crear el archivo `serverd` e inicializarlo con una configuración mínima para ejecutar `pbs_server`.

Para ver la configuración utilizar `qmgr`:





```
> pbs_server -t create
> qmgr -c 'p s'

#
# Set server attributes.
#
set server acl_hosts = kmn
set server log_events = 511
set server mail_from = adm
set server scheduler_iteration = 600
set server node_check_rate = 150
set server tcp_timeout = 6
```

Una única cola llamada 'batch' y los atributos mínimos necesarios son creados.

La configuración de SmartCloud es la siguiente:

```
#
# Create queues and set their attributes.
#
#
# Create and define queue batch
#
create queue batch
set queue batch queue_type = Execution
set queue batch resources_default.nodes = 1
set queue batch resources_default.walltime = 01:00:00
set queue batch enabled = True
set queue batch started = True
#
# Set server attributes.
#
set server scheduling = True
set server acl_hosts = server
set server default_queue = batch
set server log_events = 511
set server mail_from = adm
set server scheduler_iteration = 600
set server node_check_rate = 150
set server tcp_timeout = 300
set server job_stat_rate = 45
set server poll_jobs = True
set server mom_job_sync = True
set server next_job_number = 250
set server moab_array_compatible = True
```

6.1.6../TORQUE.SETUP





El script `torque.setup` utiliza `pbs_server -t create` para inicializar `serverdb`, y luego añade un usuario como administrador y operador de TORQUE y otros atributos necesarios. La sintaxis es `./torque.setup <username>`

```
> ./torque.setup ken
> qmgr -c 'p s'

#
# Create queues and set their attributes.
#
#
# Create and define queue batch
#
create queue batch
set queue batch queue_type = Execution
set queue batch resources_default.nodes = 1
set queue batch resources_default.walltime = 01:00:00
set queue batch enabled = True
set queue batch started = True
#
# Set server attributes.
#
set server scheduling = True
set server acl_hosts = kmn
set server managers = ken@kmn
set server operators = ken@kmn
set server default_queue = batch
set server log_events = 511
set server mail_from = adm
set server scheduler_iteration = 600
set server node_check_rate = 150
set server tcp_timeout = 6
set server mom_job_sync = True
set server keep_completed =
```

6.1.7. ESPECIFICAR NODOS DE CÓMPUTO

La variable de entorno `$TORQUEHOME` indica donde se almacenan los archivos de configuración. En esta versión de torque:

`$TORQUEHOME` es `/var/spool/torque/`.

El `pbs_server` necesita saber que elementos de la red son nodos de cómputo. Cada nodo debe ser especificado en una línea del archivo de los nodos. El archivo se encuentra en `$TORQUEHOME/server_priv/nodes`. En la mayoría de los casos, es suficiente especificar el nombre de cada nodo aun que se pueden añadir propiedades extra a cada nodo.





Sintaxis del archivo de nodos:

`node-name[:ts] [np=] [gpus=] [properties]`

La opción `[:ts]` marca el nodo como tiempo-compartido. Los nodos de tiempo-compartido son listados por el servidor en su reporte de estado pero el servidor no les asigna trabajos.

La opción `[np=]` especifica el número de procesadores virtuales de un nodo dado. Este valor puede ser menor, igual, o mayor del número de procesadores reales del nodo.

La opción `[gpus=]` especifica el número de GPUs para un nodo dado. Este valor puede ser menor, igual, o mayor del número de GPUs reales del nodo.

El número de procesadores de cada nodo puede ser detectado automáticamente por el TORQUE server si `auto_node_np` se configura como `TRUE`. Esto se puede realizar ejecutando `qmgr -c "set server auto_node_np = True"`. Configurar `auto_node_np` como `TRUE` sobrescribe el valor de `np` en:

`$(TORQUEHOME)/server_priv/nodes.`

La opción `[properties]` permite especificar cadenas de caracteres arbitrarias para identificar el nodo. Las cadenas de propiedades pueden ser alfanuméricas pero deben comenzar con una letra.

Es posible escribir comentarios en el archivo de nodos utilizando la almohadilla al comienzo de la línea (`#`).

A continuación se muestra un ejemplo del archivo de nodos

`$(TORQUEHOME)/server_priv/nodes:`

```
# Nodes 001 and 003-005 are cluster nodes
#
node001 np=2 cluster01 rackNumber22
#
# node002 will be replaced soon
node002:ts waitingToBeReplaced
# node002 will be replaced soon
#
node003 np=4 cluster01 rackNumber24
node004 cluster01 rackNumber25
node005 np=2 cluster01 rackNumber26 RAM16GB
node006
node007 np=2
node008:ts np=4
...
```





El archivo de nodos de SmartCloud tiene los siguientes nodos de ejemplo:

```
nodo1 np=2 smartcloud
nodo2 np=1 smartcloud
```

El atributo "smartcloud" debe aparecer en cada línea de configuración de los nodos, dado que se trata de una agrupación de recursos realizada en el gestor.

6.1.8. CONFIGURAR TORQUE EN LOS NODOS DE CÓMPUTO

Si se utilizan los paquetes autoextraíbles por defecto para configurar los nodos, no se necesita configuración adicional.

Si se instalan manualmente, o se necesita configuración manual adicional, editar el archivo `$TORQUEHOME/mom_priv/config` en cada nodo. Ésta es la configuración recomendada

```
$TORQUEHOME/mom_priv/config:
$pbsserver    headnode    # note: hostname running pbs_server
$logevent
```

Este archivo es idéntico para todos los nodos, puede ser creado una vez y distribuirlo.

6.1.9. FINALIZAR CONFIGURACIONES

Luego que los archivos `serverdb` y `server_priv/nodes` están configurados, y MOM tiene una configuración mínima, reiniciar `pbs_server` en el servidor y `pbs_mom` en los nodos de cómputo.

Nodos de cómputo:

```
> pbs_mom
```

Servidor:

```
> qterm -t quick
> pbs_server
```

Luego de unos segundos, el comando `pbsnodes -a` debería desplegar una lista de todos los comando con estado free.

6.1.10. CONFIGURACIÓN INICIAL DEL SERVIDOR

La configuración del demonio `pbs_server` se logra utilizando el comando `qmgr`. En un nuevo sistema, la base de datos de configuración debe ser inicializada utilizando el comando `pbs_server -t create`. Luego de esto, se debe realizar la configuración mínima para la estructura de colas y habilitar la interfaz de scheduling.





El siguiente ejemplo muestra una configuración de una implantación de una sola cola:

```
pbs_server -t create
qmgr -c "set server scheduling=true"
qmgr -c "create queue batch queue_type=execution"
qmgr -c "set queue batch started=true"
qmgr -c "set queue batch enabled=true"
qmgr -c "set queue batch resources_default.nodes=1"
qmgr -c "set queue batch resources_default.walltime=3600"
qmgr -c "set server default_queue=batch"
```

Es necesario ejecutar estos comandos como root.

En este ejemplo, la base de datos de configuración es inicializada y luego la interfaz de scheduling es activada (utilizando 'scheduling=true'). Esta interfaz permite al scheduler recibir trabajos y eventos de los nodos. El próximo paso crea una cola y especifica su tipo. Dentro de PBS, la cola debe ser declarada como 'execution' para que pueda ejecutar trabajos. Configuración adicional permite a la cola aceptar trabajos y ejecutar trabajos encolados.

Las siguientes líneas son opcionales, estableciendo el default_nodewalltime para un trabajo enviado. Estos valores van a ser utilizados en un trabajo si el mismo no especifica explícitamente otra cosa. La línea final, default_queue=batch, es una línea que indica que cualquier trabajo debe ser enviado a la cola batch a menos que se indique explícitamente que se envíe a otra cola.

Información adicional sobre la configuración puede ser encontrado en el manual de administración de qmgr y en la pagina 'man'.

Monitorizar Trabajos

TORQUE permite a usuarios y administradores monitorizar trabajos enviados con el comando qstat. Si el comando se ejecuta por un usuario no administrador, la salida solo muestra los trabajos de ese usuario, por ejemplo:

```
>qstat
```

Job id	Name	User	Time Use S Queue
4807	scatter	user01	12:56:34 R batch
...			

6.1.11. CANCELAR TRABAJOS

TORQUE permite a usuarios y administradores cancelar trabajos enviados con el comando qdel. Al trabajo se le envían las señales TERM y KILL matando todos los





procesos en ejecución. Cuando el trabajo de nivel superior termine, terminará el trabajo completo. El único parámetro es el ID del trabajo a cancelar.

Si un trabajo es cancelado por un operador, se puede configurar el envío de una notificación por correo electrónico al dueño del trabajo con la opción `-m`.

```
$ qstat
Job id      Name      User      Time Use S Queue
-----
4807       scatter  user01    12:56:34 R batch
...
$ qdel -m "hey! Stop abusing the NFS servers" 4807
$
```

6.1.12. CONFIGURACIÓN DE COLAS

Dentro de TORQUE la configuración de colas se hace con el comando `qmgr`. Con esta herramienta, el primer paso es crear una cola. Esto se logra utilizando el comando `create` dentro de la herramienta. Ejemplo:

```
>qmgr -c "create queue batch queue_type=execution"
```

Una vez creada, la cola debe ser configurada como "operacional". Como mínimo, esto incluye la asignación de atributos `started` y `enabled`. Otras configuraciones se pueden realizar con cualquier combinación de los atributos de la siguiente lista:

- Para atributos booleanos `T`, `t`, `1`, `Y`, y `are` son sinónimos de `true`, y `F`, `f`, `0`, `N`, `n` significan `false`
- EN `queue_type`, `E` y `R` son sinónimos de `Execution` y `routing`.

Configuración de cola de ejemplo

Los siguientes comandos crean y configuran una cola llamada `catch`:

```
qmgr -c "create queue batch queue_type=execution"
qmgr -c "set queue batch started=true"
qmgr -c "set queue batch enabled=true"
qmgr -c "set queue batch resources_default.nodes=1"
qmgr -c "set queue batch resources_default.walltime=3600"
```

Por defecto, un trabajo debe especificar en que cola debe ejecutarse. Para cambiar este comportamiento, se puede especificar el atributo `default_queue` como en el ejemplo:

```
qmgr -c "set server default_queue=batch"
```

TORQUE actualmente no provee un mecanismo simple para mapear colas a nodos. Sin embargo, schedulers como Moab y Maui si lo hacen.





EL método más simple es utilizar `default_resources.neednodes` en la cola de ejecución, configurándolo como un atributo del nodo.

Por ejemplo, supongamos que hay ciertos nodos pagados con dinero del departamento de química y ciertos otros con el de biología, los subconjuntos de recursos se asignan de la siguiente manera:

```
$TORQUE_HOME/server_priv/nodes:
```

```
node01 np=2 chem
```

```
node02 np=2 chem
```

```
node03 np=2 bio
```

```
node04 np=2 bio
```

```
qmgr:
```

```
set queue chem resources_default.neednodes=chem
```

```
set queue bio resources_default.neednodes=bio
```

En el caso de SmartCloud la configuración es la siguiente:

```
set queue batch resources_default.neednodes=smartcloud
```

6.2. SECURE SHELL(SSH) & AUTENTICACIÓN RSA

Abrimos la terminal y ponemos lo siguiente, que nos instalara el cliente y servidor de openssh [16]:

```
sudo apt-get install openssh-server openssh-client
```

Una vez terminado el proceso ya tenemos instalado OpenSSH Server y para probarlo, en terminal ponemos:

```
ssh localhost
```

o también y así se usa para conectarse de manera remota:
`ssh usuario@ip-de-tu-servidor`

Ahora si queremos *detener el servicio de ssh*, en la terminal ponemos:

```
sudo /etc/init.d/ssh stop
```

Para iniciarlo:

```
sudo /etc/init.d/ssh start
```

Para reiniciarlo:

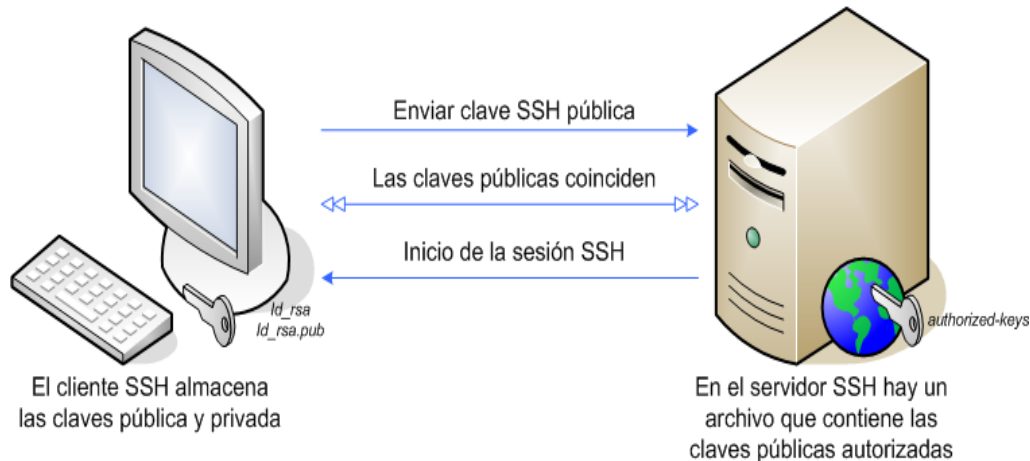
```
sudo /etc/init.d/ssh restart
```

6.2.1. IDENTIFICACIÓN SSH MEDIANTE CLAVE





SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública



Lo primero que hay que hacer es generar un conjunto de claves de cifrado asimétrico empleando el algoritmo RSA. Abrimos una terminal y escribimos lo siguiente [17]:

```
>ssh-keygen
```

A continuación nos pide que introduzcamos una *passphrase* para el conjunto de claves. La *passphrase* protege tu clave privada en caso de que se vea comprometida (ver Nota2). No obstante, en esta implementación no vamos a usar ninguna (ver Nota1 para identificación automática con *passphrase*). Posteriormente ejecutar

```
>cd .ssh/
```

Este directorio contiene el par de claves que acabamos de generar: *id_rsa* e *id_rsa.pub*. Tenemos que subir al servidor de destino la parte pública de la clave, es decir, *id_rsa.pub*.

Usamos el comando *scp* para copiar el archivo de su ubicación local a una carpeta temporal del servidor mediante SSH.

```
>scp id_rsa.pub usuario_servidor@direccion_servidor.com:/tmp
```

Conectamos al servidor mediante *ssh*.

```
>ssh usuario_servidor@direccion_servidor.com
```

Tras introducir nuestra clave nos movemos hasta este directorio:

```
cd .ssh/
```

Añadimos nuestra clave pública al listado de claves autorizadas y después borramos el archivo de clave pública de su ubicación temporal:

```
cat /tmp/id_rsa.pub >> authorized-keys
```

```
rm /tmp/id_rsa.pub
```

Cerramos la sesión SSH escribiendo *exit*. Iniciamos una nueva sesión, y si todo ha ido bien no tendremos que introducir nuestra contraseña otra vez.





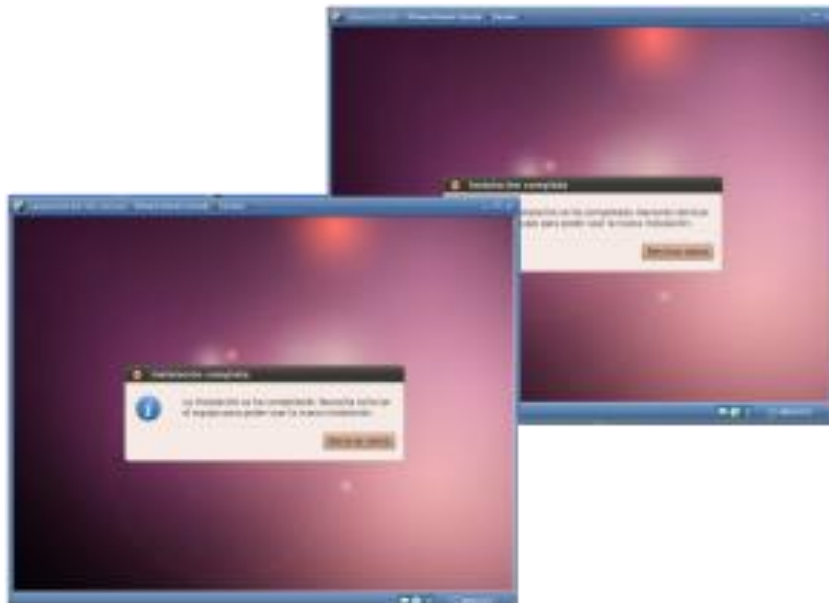
Nota1: Si queremos tener seguridad adicional utilizaremos una passphrase compleja para la parte privada de nuestra clave. Para no tener que escribir esta passphrase cada vez que nos identificamos podemos utilizar el programa ssh-agent.

Nota2: Nuestras claves se pueden ver comprometidas si alguien tiene acceso a físico a nuestro ordenador y consigue copiar los archivos de claves. También se pueden ver comprometidas si un usuario malintencionado aprovecha algún fallo de seguridad del sistema operativo para obtener privilegios de administrador y obtener los archivos de claves. Por lo tanto, hay que ponderar los riesgos y las ventajas que nos aporta la identificación automática, y tomar la decisión que más nos convenga.

6.3. NETWORK FILE SYSTEM (NFS)

6.3.1. NOTAS PRELIMINARES

Para la guía he realizado la instalación de dos máquinas virtuales con Ubuntu 10.04. Una de ellas hará las labores de server y la otra las de cliente.[19]



Servidor NFS: 192.168.2.103 con nombre 'Server'
Cliente NFS: 192.168.2.104 con nombre 'nodoN'

Todos los nodos tienen el mismo nombre, nodoN siendo n de 1 a la cantidad de nodos disponibles.

Para la configuración de los servicios se utiliza acceso por ssh, así que lo primero que hay que hacer es instalar el server de ssh en cada una de las dos máquinas.





Para ellos ejecutamos el siguiente comando tanto en el cliente como en el servidor:

```
sudo aptitude install openssh-server
```

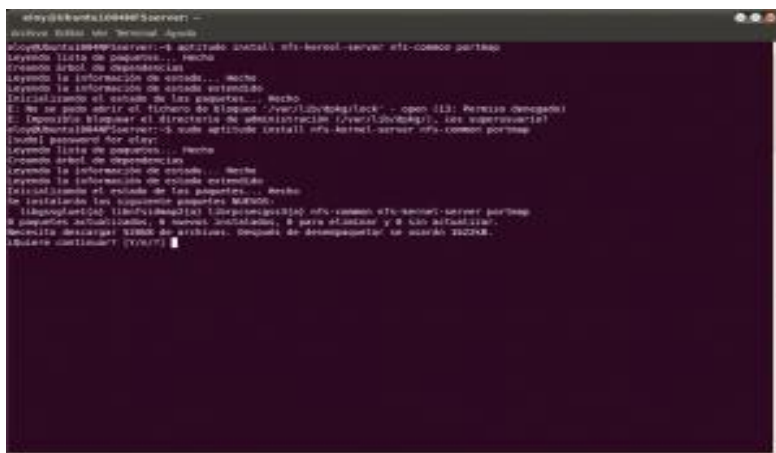
Debido a que vamos a necesitar ejecutar todos los pasos de la guía con privilegios de root, podemos o bien usar sudo delante de todos los comandos existentes en esta guía o bien pasamos a ser root en la sesión mediante:

```
sudo su
```

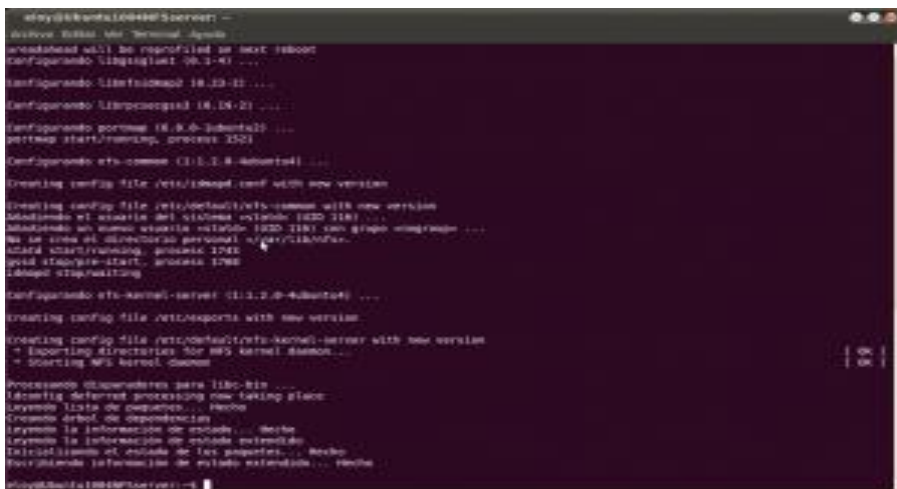
6.3.2. INSTALANDO NFS

En el servidor para la instalación ejecutamos

```
aptitude install nfs-kernel-server nfs-common portmap
```



Aquí podéis apreciar los resultados de la instalación:



En el cliente podemos instalar NFS de la siguiente forma:

```
aptitude install nfs-common portmap
```





Lo mismo que en el caso del servidor, aquí podemos ver los resultados de la instalación:

```
eIoy@Ubuntu1004: ~$ sudo aptitude install nfs-common portmap
Archivo Editar Ver Terminal Ayuda
eIoy@Ubuntu1004:~$ sudo aptitude install nfs-common portmap
[sudo] password for eIoy:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Leyendo la información de estado extendido
Inicializando el estado de los paquetes... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  libnssglue1(a) libnfsidmap2(a) librpcsecgss3(a) nfs-common portmap
0 paquetes actualizados, 5 nuevos instalados, 0 para eliminar y 0 sin actualizar.
Necesita descargar 359kB de archivos. Después de desempaquetar se usarán 1190kB.
¿Quiere continuar? [Y/n/?]
```

```
eIoy@Ubuntu1004: ~$ sudo aptitude install nfs-common portmap
Archivo Editar Ver Terminal Ayuda
Seleccionando el paquete libnfsidmap2 previamente no seleccionado.
Desempaquetando libnfsidmap2 1:0~11ubuntu1 [0] .../libnfsidmap2_1:0~11ubuntu1_amd64.deb ...
Seleccionando el paquete librpcsecgss3 previamente no seleccionado.
Desempaquetando librpcsecgss3 1:0~11ubuntu1 [0] .../librpcsecgss3_1:0~11ubuntu1_amd64.deb ...
Seleccionando el paquete portmap previamente no seleccionado.
Desempaquetando portmap 0:0~11ubuntu1 [0] .../portmap_0:0~11ubuntu1_amd64.deb ...
Seleccionando el paquete nfs-common previamente no seleccionado.
Desempaquetando nfs-common 1:0~11ubuntu1 [0] .../nfs-common_1:0~11ubuntu1_amd64.deb ...
Procesando dependencias para libnfsidmap2
portmap will be replaced as well reason
Configurando libnfsidmap2 1:0~11ubuntu1 ...
Configurando librpcsecgss3 1:0~11ubuntu1 ...
Configurando portmap 0:0~11ubuntu1 ...
portmap start/running, process 5482
Configurando nfs-common 1:0~11ubuntu1 ...
Creating config file /etc/nfsidmap.conf with new version
Creating config file /etc/default/nfs-common with new version
Sección de inicio del visor de estado [0] [0] ...
Llamando en modo monitor remoto: [0] [0] con grupo progreso ...
Se creó el directorio personal /home/eIoy/nfs/.
Se está start/running, proceso 1784
Se está stop/re-start, proceso 1729
Llamando stop/re-start
Procesando dependencias para libnfsidmap2
Llamando a otros procesos que usaban libnfsidmap2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Leyendo la información de estado extendido
Inicializando el estado de los paquetes... Hecho
Actualizando información de estado extendido... Hecho
eIoy@Ubuntu1004:~$
```

6.3.3. EXPORTANDO DIRECTORIOS EN EL SERVIDOR

Del lado del servidor, me gustaría hacer que los directorios /home y /var/nfs sean accesibles al cliente; por consiguiente debemos "exportar" esos directorios en el servidor.

Cuando un cliente accede a un recurso compartido en NFS, normalmente lo hace como el usuario "nobody". Normalmente el directorio /home no tiene como propietario ese usuario (y no recomiendo que lo tenga), y debido a que queremos que el acceso a /home sea de L/E, indicaremos en NFS que el acceso a esa ruta debe ser realizado por root (si el recurso compartido /home fuera solo de lectura, esto no sería necesario). El directorio /var/nfs no existe, así que vamos a empezar por crearlo y asociarle un propietario acorde a lo ya indicado (nobody:nogroup) de la siguiente forma:

```
mkdir /var/nfs
chown nobody:nogroup /var/nfs
```

Aquí dejamos la captura de cómo queda ese directorio ya creado y con los propietarios indicados:





```
drwxrwxrwt 2 root root 4,0K 2010-04-13 22:52 cras
drwxr-xr-x 2 root root 4,0K 2010-04-29 14:41 games
drwxr-xr-x 61 root root 4,0K 2010-10-17 02:20 lib
drwxrwsr-x 2 root staff 4,0K 2010-04-23 12:23 local
drwxrwxrwt 2 root root 60 2010-10-17 02:19 locl
drwxr-xr-x 14 root root 4,0K 2010-10-17 02:02 log
drwxrwsr-x 2 root mail 4,0K 2010-04-29 14:17 mail
drwxr-xr-x 2 nobody nogroup 4,0K 2010-10-17 02:36 nfs
drwxr-xr-x 2 root root 4,0K 2010-04-29 14:17 opt
drwxr-xr-x 16 root root 700 2010-10-17 02:20 run
drwxr-xr-x 7 root root 4,0K 2010-04-29 14:28 spool
drwxrwxrwt 2 root root 4,0K 2010-04-23 12:23 tmp
eLoy@Ubuntu1004NFSserver:/var$
eLoy@Ubuntu1004NFSserver:/var$
```

Ahora comenzamos con la configuración propiamente dicha. Vamos a modificar el fichero `/etc/exports` que es donde "exportamos" nuestros recursos compartidos mediante NFS. Vamos a indicar `/home` y `/var/nfs` como recursos compartidos NFS y vamos a decirle a NFS que el acceso a `/home` sea como root (para aprender más acerca de `/etc/exports`, su formato y opciones disponibles, podéis echar un vistazo en `man 5 exports code>`)

```
vi /etc/exports
# /etc/exports: the access control list for filesystems
which may be exported
# to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check)
hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4
gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/home 192.168.2.104(rw,sync,no_root_squash,no_subtree_check)
/var/nfs 192.168.2.104(rw,sync,no_subtree_check) code>
```

(La opción `no_root_squash` indica que `/home` deberá ser accedido mediante root). Cada vez que modifiquemos el fichero de configuración `/etc/exports`, debemos ejecutar `exportfs -a` para que los cambios tengan efecto.

Aquí tenéis las capturas de pantalla del proceso anteriormente descrito: el fichero modificado `/etc/exports`





```
eLoy@Ubuntu1004NFServer: /
Archivo Editar Ver Terminal Ayuda
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes        hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/home   gss/krb5i(rw,sync,no_subtree_check)
#
/home             192.168.2.104(rw,sync,no_root_squash,no_subtree_check)
/var/nfs          192.168.2.104(rw,sync,no_subtree_check)
/etc/exports (END)
```

La ejecución del comando exportfs:

```
eLoy@Ubuntu1004NFServer:/$
eLoy@Ubuntu1004NFServer:/$
eLoy@Ubuntu1004NFServer:/$ sudo exportfs -a
eLoy@Ubuntu1004NFServer:/$
```

Montando los recursos compartidos NFS en el cliente

Bien, lo primero que debemos hacer en el cliente es crear los directorios donde vamos a querer montar los compartidos NFS, por ejemplo:

```
mkdir -p /mnt/nfs/home
mkdir -p /mnt/nfs/var/nfs
```

Ahora del lado del cliente, podemos realizar el montaje de la siguiente forma:

```
mount 192.168.2.103:/home /mnt/nfs/home
mount 192.168.2.103:/var/nfs /mnt/nfs/var/nfs
```

```
root@Ubuntu1004:/mnt#
root@Ubuntu1004:/mnt#
root@Ubuntu1004:/mnt# mount 192.168.2.103:/home /mnt/nfs/home
root@Ubuntu1004:/mnt# mount 192.168.2.103:/var/nfs /mnt/nfs/var/nfs
root@Ubuntu1004:/mnt#
root@Ubuntu1004:/mnt#
root@Ubuntu1004:/mnt#
```

Ahora ya debíamos ser capaces de ver los dos recursos NFS en la salida del comando `df -h`

```
root@Ubuntu1004:/mnt# df -h
S.ficheros Tamaño Usado Disp Uso% Montado en
/dev/sda1 12G 2,9G 7,9G 27% /
none 496M 188K 496M 1% /dev
none 500M 80K 500M 1% /dev/shm
none 500M 108K 500M 1% /var/run
none 500M 0 500M 0% /var/lock
none 500M 0 500M 0% /lib/init/rw
192.168.2.103:/home 12G 2,9G 7,9G 27% /mnt/nfs/home
192.168.2.103:/var/nfs 12G 2,9G 7,9G 27% /mnt/nfs/var/nfs
root@Ubuntu1004:/mnt#
```

Y el comando mount retorna

```
root@Ubuntu1004:/mnt# mount
/dev/sda1 on / type ext4 (rw,errors=remount-ro)
```





```
proc on /proc type proc (rw,noexec,nosuid,nodev)
none on /sys type sysfs (rw,noexec,nosuid,nodev)
none on /sys/fs/fuse/connections type fusectl (rw)
none on /sys/kernel/debug type debugfs (rw)
none on /sys/kernel/security type securityfs (rw)
none on /dev type devtmpfs (rw,mode=0755)
none on /dev/pts type devpts
(rw,noexec,nosuid,gid=5,mode=0620)
none on /dev/shm type tmpfs (rw,nosuid,nodev)
none on /var/run type tmpfs (rw,nosuid,mode=0755)
none on /var/lock type tmpfs (rw,noexec,nosuid,nodev)
none on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc
(rw,noexec,nosuid,nodev)
192.168.2.103:/home on /mnt/nfs/home type nfs
(rw,addr=192.168.2.103)
192.168.2.103:/var/nfs on /mnt/nfs/var/nfs type nfs
(rw,addr=192.168.2.103)
```

```
root@Ubuntu1004: /mnt
Archivo Editor Ver Terminal Ayuda
root@ubuntu1004:/mnt# df -h
s.ficheros          Tamaño Usado  Disp Usado Montado en
/dev/sda1           12G  2,9G  7,9G  27% /
none                492M  180K  492M   1% /dev
none                500M   80K  500M   1% /dev/shm
none                500M  180K  500M   1% /var/run
none                500M   0  500M   0% /var/lock
none                500M   0  500M   0% /lib/init/rw
192.168.2.103:/home  12G  2,9G  7,9G  27% /mnt/nfs/home
192.168.2.103:/var/nfs  12G  2,9G  7,9G  27% /mnt/nfs/var/nfs

root@ubuntu1004:/mnt# mount
root@ubuntu1004:/mnt#
/dev/sda1 on / type ext4 (rw,errors=remount-ro)
proc on /proc type proc (rw,noexec,nosuid,nodev)
none on /sys type sysfs (rw,noexec,nosuid,nodev)
none on /sys/fs/fuse/connections type fusectl (rw)
none on /sys/kernel/debug type debugfs (rw)
none on /sys/kernel/security type securityfs (rw)
none on /dev type devtmpfs (rw,mode=0755)
none on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)
none on /dev/shm type tmpfs (rw,nosuid,nodev)
none on /var/run type tmpfs (rw,nosuid,mode=0755)
none on /var/lock type tmpfs (rw,noexec,nosuid,nodev)
none on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,noexec,nosuid,nodev)
192.168.2.103:/home on /mnt/nfs/home type nfs (rw,addr=192.168.2.103)
192.168.2.103:/var/nfs on /mnt/nfs/var/nfs type nfs (rw,addr=192.168.2.103)
root@ubuntu1004:/mnt#
```

6.4. IPYTHON

Para poder crear un perfil se tiene que poner la siguiente línea de comando por consola:

```
ipython perfil create nbserver
```

Este comando se encarga de crear todos los archivos de configuración necesarios para el iPython, en el directorio del perfil, modifique el archivo `ipython_notebook_config.py`. Por defecto, el archivo tiene todos los campos comentados, el conjunto mínimo de campos que tiene que descomentar y editar son:

```
c = get_config ()
```

Con esta instrucción se define el archivo de configuración.

```
c.NotebookApp.ip = '*'
```





Al asignarle este valor '*' permite que se pueda utilizar el notebook de forma pública, es decir poder acceder a este servicio desde otro punto de red.

`c.NotebookApp.port = 9999`

Esta línea indica el puerto donde queremos que se pueda acceder a nuestro notebook.

Para acceder al notebook con la configuración indicada sería mediante el navegador ingresar a <https://yourhost.com:9999>

6.5. APACHE WEB SERVER & MYSQL

Para instalar el servidor Web sobre Ubuntu 10.04, tendremos que instalar y configurar las siguientes aplicaciones:

- *Apache 2*: Servidor Web muy versátil y uno de los más utilizados en el mundo
- *MySQL 5*: Uno de los servidores de bases de datos más populares y utilizados en sitios y aplicaciones Web
- *PHP 5*: Un lenguaje de intérprete utilizado para construir aplicaciones y páginas web

Actualizar el sistema, siempre es una recomendación antes de llevar a cabo cualquier instalación de cualquier aplicación realizar la actualización del sistema. Lo primero es ejecutar en terminal (acude a Aplicaciones – Accesorios – Terminal) el comando:

```
sudo aptitude update  
sudo aptitude safe-upgrade
```

Con estos comandos se actualiza nuestra lista de repositorios y los paquetes que tengamos instalados si es que hay alguna actualización disponible.

6.5.1. INSTALACION Y CONFIGURACION DE APACHE 2

Apache es el servidor web más popular en el mundo, gratuito, open source, de código libre y abierto, disponible para su uso o cualquier modificación, modular, flexible y totalmente configurable. Se caracteriza por ser el principal servidor en sistemas operativos Linux, aunque también es posible instalarlo en Windows en todas sus versiones. Soporta HTTPS, virtual hosting, CGI, SSI, IPv6, scripting y integración con bases de datos, filtrado de solicitudes y respuestas, muchos esquemas de autenticación y mucho más. Para estar al tanto de las últimas





informaciones y novedades sobre este servidor Web acude a su sitio web oficial
(<http://www.apache.org>)

En primer se comprueba que no está instalado apache en el equipo. Para ello en
el terminal hay que escribir:

whereis httpd

Si la respuesta es *httpd*:significa que Apache no está instalado en el equipo

```
edita@edita-laptop:~$ whereis httpd
httpd:
edita@edita-laptop:~$
```

Si en cambio responde *httpd:/usr/local/bin/httpd* Apache si está instalado en tu
equipo y el ejecutable se encuentra en */usr/local/bin/httpd*.

Para instalarlo simplemente ejecuta en la consola o terminal:

sudo apt-get install apache2

Esto instalará Apache2 y todos los paquetes necesarios para su funcionamiento:

```
edita@edita-laptop: ~
Archivo Editar Ver Terminal Ayuda
edita@edita-laptop:~$ sudo apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
 apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common libapr1
 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
Paquetes sugeridos:
 apache2-doc apache2-suexec apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
 apache2 apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common
 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 actualizados, 9 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 3331kB de archivos.
Se utilizarán 10,1MB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]?
```

Nuestro servidor Web ya está instalado y Apache se ejecuta con normalidad:

```
Configurando apache2-mpm-worker (2.2.14-5ubuntu8.4) ...
* Starting web server apache2
apache2: Could not reliably determine the server's fully
using 127.0.1.1 for ServerName

Configurando apache2 (2.2.14-5ubuntu8.4) ...

Procesando disparadores para libc-bin ...
ldconfig deferred processing now taking place
edita@edita-laptop:~$
```

Una vez finalizada la instalación, puedes comprobar la versión que has instalado
de Apache ejecutando en la consola

apache2 -v





Para probar en modo gráfico que, en efecto, Apache está en funcionamiento acude a Firefox e introduce en el campo Dirección del navegador *http://localhost* o bien *http://127.0.0.1*

Debe aparecer una página específica indicándote que Apache se está ejecutando:



It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

Para manejar el servidor Apache, estos son los comandos más básicos:

- Iniciar Apache2 – `sudo /etc/init.d/apache2 start` o bien `sudo service apache2 start`
- Detener Apache2 – `sudo /etc/init.d/apache2 stop` o bien `sudo service apache2 stop`
- Reiniciar Apache2 – `sudo /etc/init.d/apache2 restart` o bien `sudo service apache2 restart`
- Estatus Apache – `sudo /etc/init.d/apache2 status` o bien `sudo service apache2 status`

El archivo de configuración por defecto de Apache se encuentra en el directorio `/etc/apache2/` y el directorio por defecto de los archivos donde se guardarán las páginas web es en `/var/www`. (Por ejemplo, para instalar un CMS como Joomla o Drupal, un gestor de contenidos como Wordpress o la aplicación educativa Moodle, tendrás que ubicar en esta carpeta todos los archivos necesarios para acometer la instalación: cada aplicación en su carpeta específica, claro, debes ser ordenado).

6.5.2. INSTALAR MYSQL

MySQL es el sistema gestor de bases de datos que instalaremos en nuestro servidor. Posee dos licencias en función del uso al que va a ser sometido: licencia GPL si lo utilizamos con software libre o privativo si se desea incorporar dentro de un software privativo o se desea comprar. Es utilizado por Google por ejemplo, para su motor de búsqueda, por la Wikipedia, Flickr, etc. Lo que nos indica que tiene gran estabilidad, fiabilidad, robusto, flexible, fácil de configurar y suficientemente rápido para la mayoría de las aplicaciones comunes, siendo especialmente eficaz en cuanto a aplicaciones web se refiere.



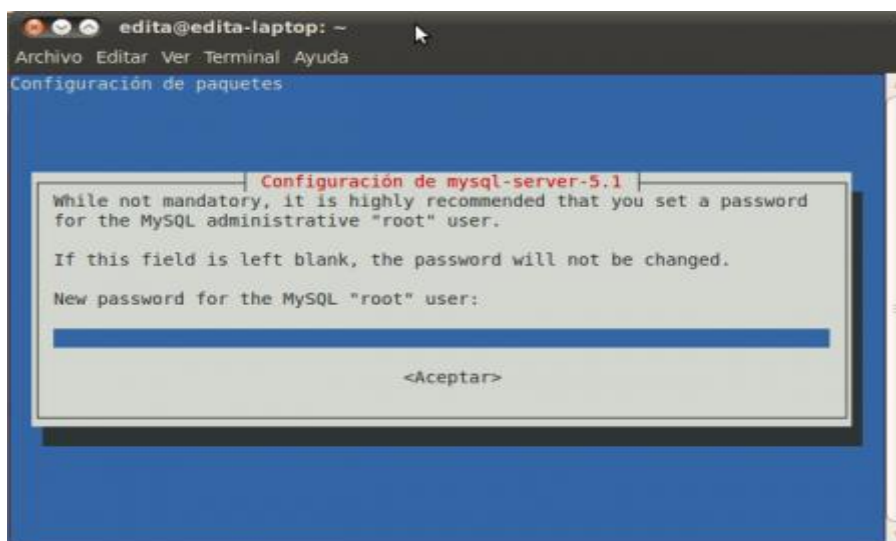


Para instalarlo ejecuta el siguiente comando:
`sudo apt-get install mysql-server`

Esto instalará todos los paquetes y librerías necesarias para que funcione el servidor MySQL.

```
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
 libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient16
 libnet-daemon-perl libplrpc-perl mysql-client-5.1 mysql-client-core-5.1
 mysql-common mysql-server-5.1 mysql-server-core-5.1
Paquetes sugeridos:
 dbshell libipc-sharedcache-perl tinyca mailx
Se instalarán los siguientes paquetes NUEVOS:
 libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient16
 libnet-daemon-perl libplrpc-perl mysql-client-5.1 mysql-client-core-5.1
 mysql-common mysql-server mysql-server-5.1 mysql-server-core-5.1
0 actualizados, 12 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 23,3MB de archivos.
Se utilizarán 54,8MB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]?
```

Durante la instalación aparece una pantalla solicitando la clave de root para el servidor MySQL. En esta ventana tendrás que escribir una clave para el administrador de la base de datos (no es recomendable que sea la misma clave del usuario de Ubuntu), debe tener una longitud mayor de 8 caracteres y mezclar letras, símbolos y números siempre es una buena opción. No están permitidos los espacios en blanco.



Una vez finalice, MySQL está instalado y en funcionamiento:





```
Configurando mysql-server-5.1 (5.1.41-3ubuntu12.10) ...  
mysql start/running, process 18005  
... anterior y aparecerá el prompt mysql> algo parecido  
Configurando libhtml-template-perl (2.9-1) ...  
Configurando mysql-server (5.1.41-3ubuntu12.10) ...  
Procesando disparadores para libc-bin ...  
ldconfig deferred processing now taking place  
edita@edita-laptop:~$
```

Para comprobar el funcionamiento del gestor MySQL ejecutar el siguiente comando:

```
mysql -u root -p
```

Se solicitará la clave de usuario root de MySQL (la que se registro en la ventana anterior). Tras escribirla aparecerá el prompt mysql>

```
edita@edita-laptop:~$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \q  
Your MySQL connection id is 39  
Server version: 5.1.41-3ubuntu12.10 (Ubuntu)  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql>
```

Los principales comandos para administrar MySQL son:

- Iniciar MySQL – sudo /etc/init.d/mysql start o bien sudo service mysql start
- Detener MySQL – sudo /etc/init.d/mysql stop o bien sudo service mysql stop
- Reiniciar MySQL – sudo /etc/init.d/mysql restart o bien sudo service mysql restart
- Estatus MySQL – sudo /etc/init.d/mysql status o bien sudo service mysql status

El archivo de configuración de mysql se localiza en /etc/mysql/my.cnf. Si se modifica este archivo se tiene que reiniciar el servicio MySQL.

Para por medio de scripts remotos (es decir, no albergados en tu propio servidor) debes editar el bind-address en /etc/mysql/my.cnf y reemplazar el valor que trae por defecto (127.0.0.1) por tu dirección IP.

6.5.3. INSTALAR PHP

PHP es un lenguaje de intérprete que puede ser usado principalmente para tres tareas: escribir páginas dinámicas, crear programas de consola o mediante el uso





conjunto de GTK codificar aplicaciones gráficas. Es un lenguaje de programación web del lado del servidor. El cliente realiza la petición al servidor, éste la procesa con sus scripts PHP y devuelve una salida al cliente que no incluye nada de PHP. En el caso de otros lenguajes de programación dinámica de web como Javascript, éste se devuelve al cliente y se ejecuta en él. Tecnologías similares a PHP son ASP de Microsoft, ColdFusion de Adobe, ISP/Java de Sun Microsystems y CGI/Perl. Las ventajas de PHP son su facilidad de uso, su carácter de software libre, su capacidad de conexión con la mayoría de los sistemas gestores de bases de datos, la profusa documentación disponible en la red y su capacidad de expansión mediante el uso de módulos. La gran mayoría de CMS libres están programados utilizando esta tecnología. De entre las plataformas que usan PHP se encuentran Wordpress, Drupal o el mismo Joomla.

Para instalar PHP 5, junto con el soporte para Apache y MySQL, escribe el siguiente comando en la consola o terminal:

sudo apt-get install php5 libapache2-mod-php5 php5-mysql

```
edita@edita-laptop:~$ sudo apt-get install php5 libapache2-mod-php5 php5-mysql
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
 apache2-mpm-prefork php5-common
Paquetes sugeridos:
 php-pear php5-suhosin
Los siguientes paquetes se ELIMINARÁN:
 apache2-mpm-worker
Se instalarán los siguientes paquetes NUEVOS:
 apache2-mpm-prefork libapache2-mod-php5 php5 php5-common php5-mysql
0 actualizados, 5 se instalarán, 1 para eliminar y 0 no actualizados.
Necesito descargar 3449kB de archivos.
Se utilizarán 9048kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]?
```

Una vez finalizado:

```
Creating config file /etc/php5/apache2/php.ini with default values
* Reloading web server config apache2
apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.1.1 for ServerName

Configurando php5 (5.3.2-1ubuntu4.9) ...
Configurando php5-mysql (5.3.2-1ubuntu4.9) ...
edita@edita-laptop:~$
```

Ahora es necesario reiniciar el servidor Apache para que tome los cambios de la librería de PHP. Si no lo haces no funcionará PHP hasta que no reinicies el servidor Apache. Por ello en la consola o terminal escribe el siguiente comando:

sudo /etc/init.d/apache2 restart

Para probar que todo va bien y que PHP funciona perfectamente crea un archivo de nombre test.php con tu editor favorito, por ejemplo, gedit que incluya un





SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

fragmento de código PHP que pueda ser interpretado por dicho lenguaje de interpretación.

En la consola o terminal escribe:

```
sudo gedit /var/www/test.php
```

Se abrirá así el editor de textos Gedit. Escribe en él lo siguiente:

```
<?php phpinfo(); ?>
```



Guarda el archivo con el texto anterior escrito. Acude a continuación a Firefox y escribe la dirección <http://localhost/test.php> y observa cómo se despliega una lista con toda la información de PHP, los módulos y librerías instalados.

PHP Version 5.3.2-1ubuntu4.9	
System	Linux edItA-laptop 10.32-32-generic #62-Ubuntu SMP Wed Apr 20 21:54:21 UTC 2011; root@edItA
Build Date	May 3 2011 00:31:05
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini
PHP API	20090626
PHP Extension	20090626
Zend Extension	220090626

Si aparece dicha pantalla es que PHP está correctamente instalado y funcionando perfectamente con Apache.

El archivo de configuración de PHP se encuentra en `/etc/php5/apache2/php.ini`





7. ESTUDIO DE COSTES E IMPACTO ECONÓMICO

Uno de los objetivos de SmartCloud es lograr optimizar la utilización de la infraestructura existente, lo que se refleja en un ahorro en costes. Actualmente si la universidad necesita realizar cálculos que exceden sus capacidades de procesamiento debe recurrir a un proveedor externo [5]. Al utilizar el cluster dinámico de SmartCloud, se pueden ofrecer los mismos servicios de procesamiento sin tener que contratar a un proveedor externo y sin la latencia que implica el procesamiento remoto. El único sobrecoste significativo que implica esta plataforma, es el coste energético [10]. A continuación se detalla el modelo económico desarrollado para realizar estos cálculos [11].

$$C = W_i * T * F_c * F_u * C_e$$

$$O = \frac{W_i * (T_b + F_c * F_u * T_i) + W_i * (T_i * (1 - F_c * F_u))}{W_i * T_b + W_i * T_i}$$

En donde términos de costes C es el coste energético final y O el sobrecoste energético derivado de utilizar SmartCloud. W_i es el consumo energético de cada nodo con una carga estándar en Kw/h y W_i el consumo energético de cada nodo estado inactivo en Kw/h. T se refiere al tiempo disponible para el procesamiento, T_b al Tiempo de utilización del cluster con carga sin utilizar SmartCloud, y T_i al tiempo de utilización del cluster en estado inactivo sin utilizar SmartCloud. Para estimar la carga se usan las variables F_c como factor de carga del cluster en el tiempo disponible, y F_u como factor de utilización de las horas disponibles. El coste de la energía en Kw/h se expresa con C_e .

Se plantea un escenario de ejemplo para estudiar el problema. Se dispone de 20 ordenadores i7-3770K con 4GB de RAM y 4 cores con hyper-threading, en un laboratorio con la distribución de uso indicada en la figura.





SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

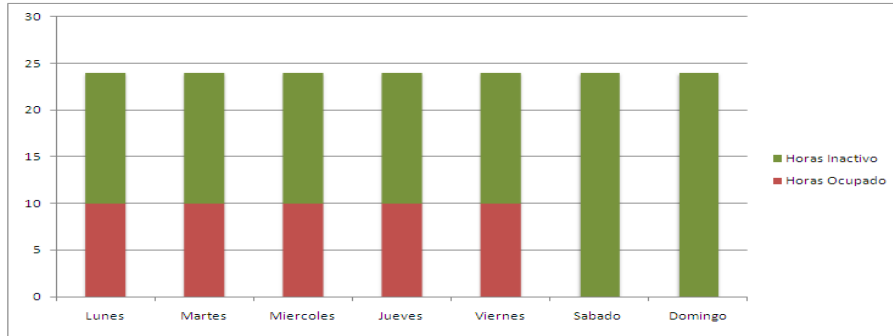


Ilustración 9 Distribución de uso de uso de ordenadores en el caso de estudio.

Datos del escenario planteado:

Día	Horas Ocupado	Horas Inactivo
Martes	10	14
Jueves	10	14
Viernes	10	14
Sábado	0	24
Domingo	0	24
	50	118

Supongo Core i7-3770K - 4Cores /8 Threads - 3.5 Ghz

Consumo Idle	84	Watt
Consumo Carga	200	Watt

Numero Maquinas	20
Cores / Maquina	8 (4 con Hyper Threading)
Ram/Maquina	4
Total Cores	160
Total Ram	80 Gb

Consumo Semanal Actual (KWh)	Coste/Semana	Coste/Mes
Inactivo:	9,912	1,03 €
Carga:	10,00000	1,04 €
		8,30 €

Porcentaje Horas inactivas Semanales ejecutando SmartCloud	100%
Factor de Carga durante uso Smart Cloud	100%

Consumo Semanal Actual (KW/h)	Coste/Semana	Coste/Mes
Inactivo:	0	- €
Carga:	33,60000	3,50 €
		14,00 €

El precio del KW/h en España es: 0.104€ [18]





SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

En una semana hay 118 horas inactivas que podrían ser utilizadas por SmartCloud, formando un cluster de 160 cores con 80 GB de RAM.

Supongamos que se necesita utilizar todo el clúster al 100 % de la carga, en el total de las horas disponibles. El consumo energético sería de 196,6€ lo que presenta un sobrecoste de aproximadamente un 68 % respecto al consumo de energía actual. Ahora bien, si se recurriera a proveedores externos para realizar este procesamiento, esto tendría un coste de media de 1508,68€. El cuadro I muestra una comparación de proveedores de servicios cloud que proporcionan un servicio equivalente.

Para calcular el ahorro final simplemente basta con restar al coste externo el coste energético, lo que se refleja en 1312€ de ahorro. En términos de latencias, el trabajo dentro de una red local como lo que propone SmartCloud implica latencias sensiblemente menores frente a las que se experimentan trabajando con proveedores cloud. El cuadro II muestra las latencias de Amazon desde España.

Claramente son mayores a las latencias dentro de una red local. A su vez, iPython es código abierto, lo que comparado con herramientas como Matlab o Mathematica que tienen costes de licencias representa un ahorro significativo.

Tipo de Servidor	CPU	RAM	HDD	Precio por hora (USD)	Precio por hora (EUR)	Coste Mensual
HP	04xCPU	4GB	120.0GB	\$ 0.14	0,108 €	1.021,597 €
RackSpace	04xCPU	4GB	100.0GB	\$ 0.22	0,168 €	1.583,475 €
SoftLayer	06x1.1GHz	7.5G B	0.0GB	\$ 0.23	0,178 €	1.678,338 €
AWS	04x1.0GHz	7.5G B	850.0GB	\$ 0.24	0,186 €	1.751,309 €

Tabla 10 Comparación de proveedores Cloud [13]

Servicios	Tiempo (seg)	#	Min ms	Max ms	Std Dev m	Media ms	Promedio
Amazon EC2	1.13	3	231	298	13.27 %	288	272.33
Amazon	0.67	6	62	94	15.69 %	78	75.5

Tabla 11 Latencias de Amazon desde España. [14]





8. CASOS DE USO

8.1. REGISTRO EN SMARTCLOUD

Nombre: Registro en SmartCloud		
Descripción	El usuario utiliza SmartCloud por primera vez y debe darse de alta en el sistema	
Objetivo	Registrarse en la aplicación	
Precondición	El usuario no debe tener un usuario en la aplicación	
Actor Principal	Usuario	
Canal del actor principal	Interfaz Web	
Disparador	El usuario presiona el botón "Registro" en la interfaz web	
Secuencia	Paso Acción	
	1	El usuario accede a la aplicación web
	2	El usuario presiona el botón "Registro" en la interfaz web
	3	El usuario introduce sus datos personales y presiona "Aceptar"
	4	El usuario selecciona si desea utilizar autenticación con Google Drive (opcional)
	4.1	Si se da el paso 4, El usuario debe autorizar a SmartCloud a utilizar Google ID (validación en servidor de Google)
	5	El sistema crea un usuario con los datos ingresados y lo guarda en la base de datos
6	El sistema notifica al usuario de la creación del usuario	
Poscondición	Se crea un nuevo usuario en el sistema	
Excepciones	4	El usuario ya existe en el sistema, el usuario es notificado y se vuelve al paso 4.
	5	Error de conexión. El usuario es





	notificado y se vuelve al paso 4
Frecuencia de uso	Habitual
Importancia	Alta
Estabilidad	Alta
Comentarios	El identificador del usuario en el sistema es su dirección de correo

8.2. LOGIN CON GOOGLE ID

Nombre: Login sin Google Drive													
Descripción	El usuario accede a la aplicación con su cuenta de Google asociada, lo que le habilita las funcionalidades de Google Drive (sincronización de archivos y login integrado)												
Objetivo	Acceder a SmartCloud con las funcionalidades de Google Drive habilitadas												
Precondición	El usuario debe tener un usuario en la aplicación asociado con su cuenta de Google ID y haber autorizado a SmartCloud												
Actor Principal	Usuario												
Canal del actor principal	Interfaz Web												
Disparador	El usuario presiona el botón "Acceder con Google ID" en la interfaz web												
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El usuario accede a la aplicación web</td> </tr> <tr> <td>2</td> <td>El usuario presiona el botón "Acceder con Google ID" en la interfaz web</td> </tr> <tr> <td>3</td> <td>El usuario es redirigido al login de Google</td> </tr> <tr> <td>4</td> <td>El usuario ingresa su Id de Google y Contraseña</td> </tr> <tr> <td>5</td> <td>El usuario accede a la aplicación y es redirigido a la interfaz de Ipython con sus archivos personales y la pestaña de "Google Drive" activada</td> </tr> </tbody> </table>	Paso	Acción	1	El usuario accede a la aplicación web	2	El usuario presiona el botón "Acceder con Google ID" en la interfaz web	3	El usuario es redirigido al login de Google	4	El usuario ingresa su Id de Google y Contraseña	5	El usuario accede a la aplicación y es redirigido a la interfaz de Ipython con sus archivos personales y la pestaña de "Google Drive" activada
Paso	Acción												
1	El usuario accede a la aplicación web												
2	El usuario presiona el botón "Acceder con Google ID" en la interfaz web												
3	El usuario es redirigido al login de Google												
4	El usuario ingresa su Id de Google y Contraseña												
5	El usuario accede a la aplicación y es redirigido a la interfaz de Ipython con sus archivos personales y la pestaña de "Google Drive" activada												
Poscondición	Se crea una nueva instancia de iPython												
Excepciones	<table border="1"> <tbody> <tr> <td>4</td> <td>Error de autenticación, vuelve al paso 4</td> </tr> <tr> <td>5</td> <td>Google Drive no está disponible, la pestaña correspondiente está deshabilitada</td> </tr> </tbody> </table>	4	Error de autenticación, vuelve al paso 4	5	Google Drive no está disponible, la pestaña correspondiente está deshabilitada								
4	Error de autenticación, vuelve al paso 4												
5	Google Drive no está disponible, la pestaña correspondiente está deshabilitada												
Frecuencia de uso	Habitual												
Importancia	Alta												
Estabilidad	Alta												





Comentarios	La disponibilidad de esta funcionalidad depende de la conexión y el estado de Google.
--------------------	---

8.3. LOGIN SIN GOOGLE ID

Nombre: Login sin Google Drive													
Descripción	El usuario accede a la aplicación sin su cuenta de Google asociada, lo que le no habilita las funcionalidades de Google Drive (sincronización de archivos y login integrado)												
Objetivo	Acceder a SmartCloud SIN las funcionalidades de Google Drive habilitadas												
Precondición	El usuario debe tener un usuario en la aplicación sin cuenta de Google ID asociada												
Actor Principal	Usuario												
Canal del actor principal	Interfaz Web												
Disparador	El usuario presiona el botón "Acceder sin Google ID" en la interfaz web												
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El usuario accede a la aplicación web</td> </tr> <tr> <td>2</td> <td>El usuario presiona el botón "Acceder sin Google ID" en la interfaz web</td> </tr> <tr> <td>3</td> <td>El usuario es redirigido al login propio de SmartCloud</td> </tr> <tr> <td>4</td> <td>El usuario ingresa su email y su contraseña</td> </tr> <tr> <td>5</td> <td>El usuario accede a la aplicación y es redirigido a la interfaz de Ipython con sus archivos personales y la pestaña de "Google Drive" desactivada</td> </tr> </tbody> </table>	Paso	Acción	1	El usuario accede a la aplicación web	2	El usuario presiona el botón "Acceder sin Google ID" en la interfaz web	3	El usuario es redirigido al login propio de SmartCloud	4	El usuario ingresa su email y su contraseña	5	El usuario accede a la aplicación y es redirigido a la interfaz de Ipython con sus archivos personales y la pestaña de "Google Drive" desactivada
Paso	Acción												
1	El usuario accede a la aplicación web												
2	El usuario presiona el botón "Acceder sin Google ID" en la interfaz web												
3	El usuario es redirigido al login propio de SmartCloud												
4	El usuario ingresa su email y su contraseña												
5	El usuario accede a la aplicación y es redirigido a la interfaz de Ipython con sus archivos personales y la pestaña de "Google Drive" desactivada												
Poscondición	Se crea una nueva instancia de iPython												
Excepciones	<table border="1"> <tbody> <tr> <td>4</td> <td>Error de autenticación, vuelve al paso 4</td> </tr> </tbody> </table>	4	Error de autenticación, vuelve al paso 4										
4	Error de autenticación, vuelve al paso 4												
Frecuencia de uso	Habitual												
Importancia	Alta												
Estabilidad	Alta												
Comentarios	-												

8.4. UTILIZAR IPYTHON PARA GENERAR/UTILIZAR NOTEBOOK

Nombre:	Utilizar iPython para Generar/Utilizar Notebook
----------------	--





Descripción	El usuario crea o abre notebook existente para realizar cálculos	
Objetivo	Utilizar la aplicación	
Precondición	Usuario loggeado en el sistema	
Actor Principal	Usuario	
Canal del actor principal	Interfaz Web	
Disparador	El usuario abre un notebook	
Secuencia	Paso	Acción
	1	El usuario abre un notebook y hace cambios
	2	El usuario presiona el botón "play"
	3	El usuario presiona guardar (opcional)
	3.1	Si se cumple la opción 3 el sistema guarda los cambios
4	El usuario cierra la aplicación o vuelve al paso 1	
Poscondición	(Opcionalmente) se guardan los cambios	
Excepciones	2	Ocurre algún error, el iPython Kernel despliega un mensaje
Frecuencia de uso	Habitual	
Importancia	Alta	
Estabilidad	Alta	
Comentarios	-	

8.5. UTILIZAR EL CLUSTER (INCIAR Y DETENER)

	Nombre: Utilizar el Cluster (iniciar y detener)	
Descripción	El usuario necesita realizar procesamiento en el cluster por lo que lo inicia dentro de iPython. El cluster utilizado es la infraestructura generada por smartcloud.	
Objetivo	Iniciar y detener cluster	
Precondición	Usuario loggeado en el sistema	
Actor Principal	Usuario	
Canal del actor principal	Interfaz Web	
Disparador	El usuario inicia el cluster 'SmartCloud' en la pestaña Clusters	
Secuencia	Paso	Acción
	1	El usuario accede a la pestaña Clusters
	2	El usuario ingresa el numero de procesadores en el cluster 'SmartCloud'





	3	El usuario presiona 'Start'
	4	Se procesa la información
	5	El usuario detiene el cluster
Poscondición	Al detener el cluster hay recursos disponibles para otro usuario.	
Excepciones	3	Si hay un error al iniciar el cluster,el sistema despliega un mensaje. Se vuelve al paso 3
Frecuencia de uso	Variable	
Importancia	Alta	
Estabilidad	Alta	
Comentarios	-	

8.6. SINCRONIZAR ARCHIVOS CON GOOGLE DRIVE

	Nombre: Sincronizar archivos con Google Drive	
Descripción	El usuario tiene archivos en local y en google drive. Se realiza un procedimiento de sincronización	
Objetivo	Mantener archivos sincronizados	
Precondición	Usuario loggeado en el sistema y con una cuenta de google id configurada y habilitada	
Actor Principal	Usuario	
Canal del actor principal	Interfaz Web	
Disparador	El usuario abre la pestaña 'Google Drive' en iPython. Ahí o bien presiona 'Sincronizar todos' o 'Sincronizar un archivo específico'	
Secuencia	Paso	Acción
	1	El usuario accede a la pestaña Google drive
	2	El usuario presiona 'Sincronizar todos' o 'Sincronizar un archivo específico'
	3	El sistema sincroniza los archivos
	4	El sistema notifica cuando estén sincronizados los archivos seleccionados
Poscondición	(Opcionalmente) se guardan los cambios	
Excepciones	3	Si hay un error al sincronizar los archivos,el sistema despliega un mensaje. Se vuelve al paso 3
Frecuencia de uso	Variable	
Importancia	Alta	
Estabilidad	Alta	





Comentarios

Depende de el estado de los servidores de Google Drive

8.7. CONSULTAR ESTADO DEL CLÚSTER

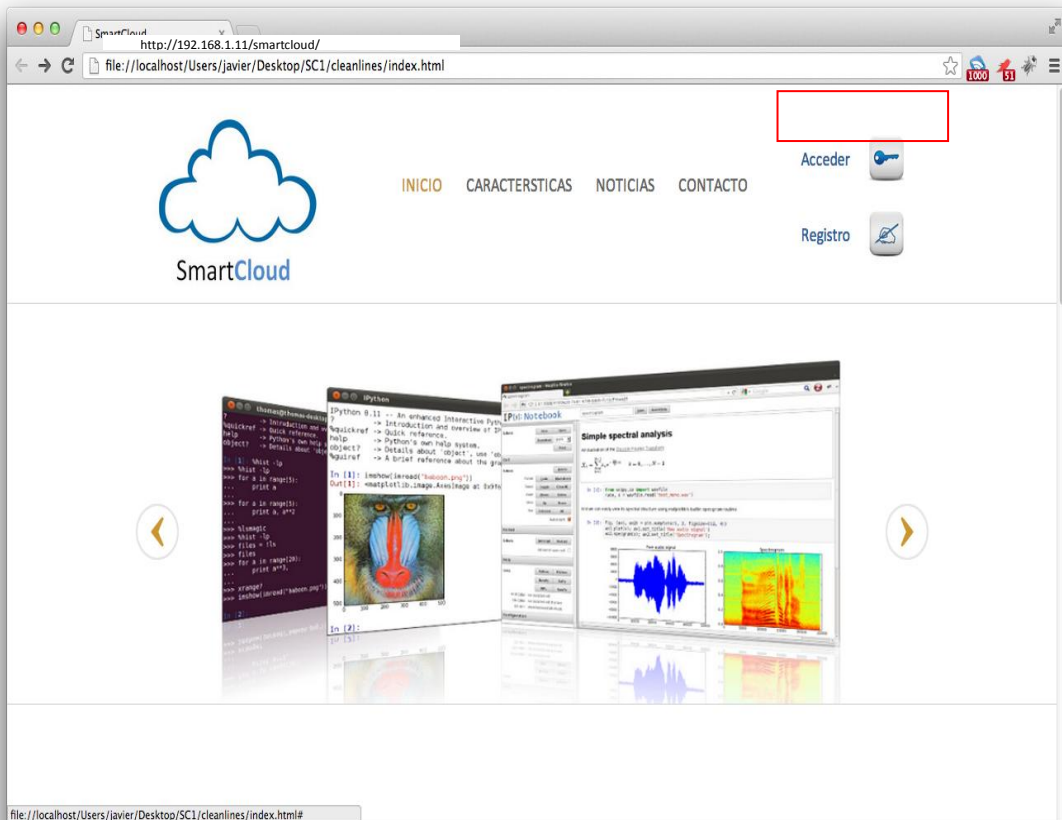
Nombre: Consultar el estado del cluster									
Descripción	El usuario puede consultar dentro de iPython el estado del cluster en términos de recursos								
Objetivo	Obtener información del cluster								
Precondición	Usuario loggeado en el								
Actor Principal	Usuario								
Canal del actor principal	Interfaz Web								
Disparador	El usuario abre la pestaña 'SmartCloud Cluster information' en iPython. Ahí presiona actualizar								
Secuencia	<table border="1"><thead><tr><th>Paso</th><th>Acción</th></tr></thead><tbody><tr><td>1</td><td>El usuario accede a la pestaña SmartCloud cluster information</td></tr><tr><td>2</td><td>El usuario presiona 'Actualizar'</td></tr><tr><td>3</td><td>El sistema despliega información del cluster</td></tr></tbody></table>	Paso	Acción	1	El usuario accede a la pestaña SmartCloud cluster information	2	El usuario presiona 'Actualizar'	3	El sistema despliega información del cluster
	Paso	Acción							
	1	El usuario accede a la pestaña SmartCloud cluster information							
	2	El usuario presiona 'Actualizar'							
3	El sistema despliega información del cluster								
Poscondición	(Opcionalmente) se guardan los cambios								
Excepciones	-								
Frecuencia de uso	Variable								
Importancia	Media								
Estabilidad	Alta								
Comentarios									





9. INTERFAZ DEL SISTEMA

9.1. PÁGINA DE INICIO





9.2. FORMULARIO DE REGISTRO

The screenshot shows a web browser window with the URL `http://192.168.1.11/smartcloud/Registro.html`. The page features the SmartCloud logo and navigation links: INICIO, CARACTERISTICAS, NOTICIAS, and CONTACTO. There are buttons for 'Acceder' and 'Registro'. The registration form includes fields for 'Username' (containing 'prueba') and 'Password' (masked with dots). Below these fields is a checkbox for 'Vincular con Google Drive?' with radio buttons for 'Si' and 'No'. At the bottom of the form are 'Aceptar' and 'Cancelar' buttons.

9.3. NOTEBOOK

9.3.1. LISTADOS DE NOTEBOOKS

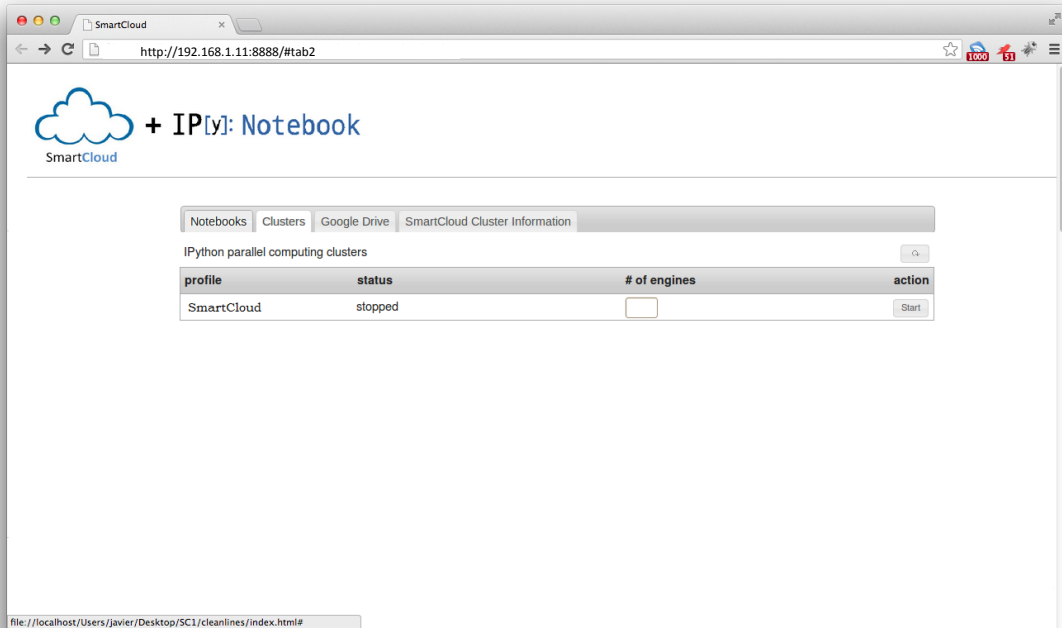
The screenshot shows a web browser window with the URL `http://192.168.1.11:8888`. The page displays the SmartCloud logo and the text '+ IP[y]: Notebook'. Below this, there are tabs for 'Notebooks', 'Clusters', 'Google Drive', and 'SmartCloud Cluster Information'. A message states: 'To import a notebook, drag the file onto the listing below or [click here](#).' There is a 'New Notebook' button. A table lists several notebooks with their names and 'Delete' buttons:

Notebook Name	Action
/home/server/Escritorio	
Animations_and_Progress	Delete
Model Selection for the Nystroem Method	Delete
Parallel Magics	Delete
Untitled0	Delete
Untitled1	Delete
XKCD_plots	Delete
helloworld	Delete
ipy-parallel-push-exec-pull	Delete
just_plot_it_blog_post_part2	Delete
taskmap	Delete

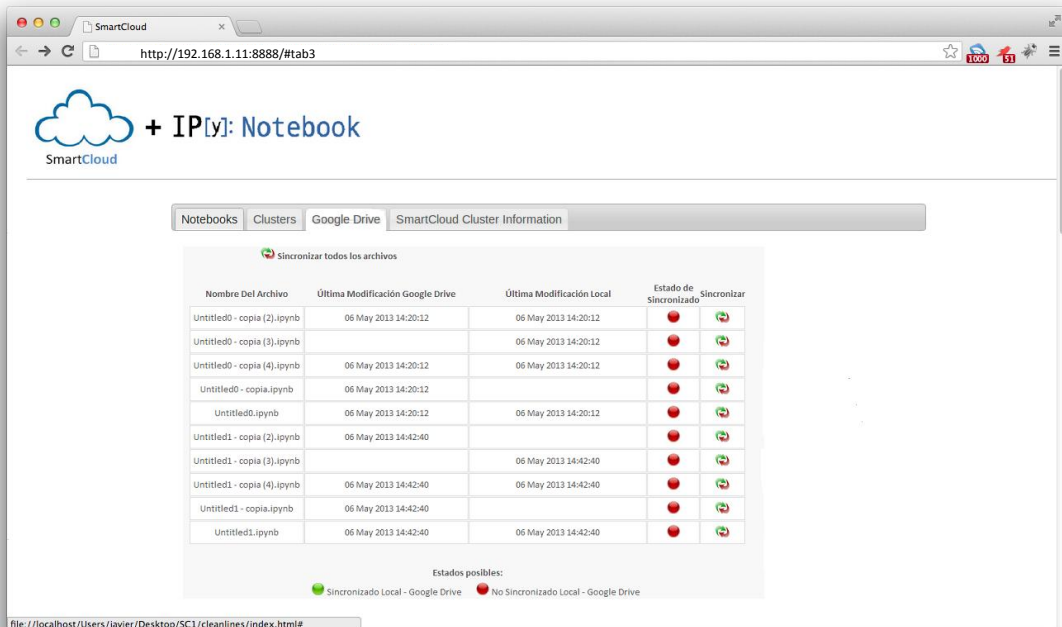




9.3.2. CLÚSTER



9.3.3. GOOGLE DRIVE





9.3.4. INFORMACIÓN DEL CLÚSTER

The screenshot shows a web browser window with the URL `http://192.168.1.11:8888/#tab4`. The page title is "SmartCloud + IP[y]: Notebook". Below the title, there are navigation tabs: "Notebooks", "Clusters", "Google Drive", and "SmartCloud Cluster Information". The "SmartCloud Cluster Information" tab is active, showing a table with the following data:

Nodo	Num Procesadores	Estado	Jobs
nodo1	4	down	1/249[1_server, 2/249[2_server, 3/249[3_server
nodo2	1	down	

At the top of the table, there is a button labeled "Actualizar" and a text indicating the last update: "Ultima Actualización: 2013-05-12 17:06:43".

9.3.5. NOTEBOOK

The screenshot shows a web browser window with the URL `http://192.168.1.11:8888/adfasd-asdad-asdadadsds`. The page title is "SmartCloud + IP[y]: Notebook". Below the title, there is a menu bar with options: "File", "Edit", "View", "Insert", "Cell", "Kernel", and "Help". Below the menu bar, there is a toolbar with various icons and a dropdown menu set to "Code". The main content area is empty, with a prompt "In []:" at the top.





10. CONTRIBUCIONES Y TRABAJOS FUTUROS

La importancia que tiene la utilización de herramientas que utilizan el cloud se reflejan en el ahorro de costes tanto energéticos como en infraestructura tal como se refleja en el actual trabajo.

Uno de los principales aportes de este trabajo es la posibilidad de poder utilizarlo de manera multiusuario, ya que inicialmente la herramienta iPython estaba diseñada para soportar dicho propósito pero no implementada.

Se ofrece a los investigadores no sólo una poderosa herramienta de cálculo sino también una forma de aprovechar los recursos que existen sin grandes costes añadidos, además de estar integrado con Google Drive lo que facilita mucho más su usabilidad.

El proyecto desplegado sobre una arquitectura en cluster, también implementada, permite el despliegue de otras herramientas y recursos como apoyo a otros proyectos, lo que facilita la integración con cualquier aplicación paralelizable.

Este proyecto puede continuar para seguir explotando la escalabilidad e integración de otros recursos. Se puede dar la posibilidad de integrarlo con otros servicios cloud por ejemplo los de Amazon [27], el diseño se puede realizar desde distintos enfoques, pues se adapta a las tendencias actuales de Cloud, permitiendo una mayor versatilidad. [28]

Para este proyecto desplegamos en el cluster una herramienta en concreto, iPython, pero cualquier aplicación paralelizable podría ser desplegada, por ejemplo aplicaciones más complejas o específicas de un proyecto o de alguna empresa que en determinado momento tenga un pico alto de consumo de recursos.





SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

Una vez logrado el principal objetivo y consolidado la infraestructura, se podría ofrecer estos servicios de cálculo a otras facultades o entidades externas.

Un estudio a profundidad en lo referente a seguridad permitirá resolver los posibles fallos a los que puede estar expuesto, en el actual desarrollo solo se cubren los estándares mínimos necesarios.

Se puede trabajar en un modelo de cluster híbrido con máquinas virtuales, servidores físicos y con balanceadores de carga como alternativa a la actual implementación. También se puede levantar otras imágenes sobre las vm para dar instancias únicas.





11. REPERCUSIÓN

El proyecto SmartCloud tiene gran repercusión en los medios. Para más información consultar los anexos.

1. Artículo en HPC In the Cloud

URL: http://www.hpcinthecloud.com/hpccloud/2013-01-04/student_projects_highlight_cloud_s_potential.html

2. Entrevista en la radio Cadena SER

El martes 15 de enero fuimos entrevistados en la cadena SER para explicar el proyecto de fin de carrera y el impacto que podría tener como ganancia para la Universidad.

El audio se puede oír en:

URL: http://www.cadenaser.com/cultura/audios/noticiasGmadridGcharlamosGuniversidad/csrcsrpor/20130115csrsrcul_8/Aes/

3. Artículo en Tribuna Complutense

4. Artículo en la Facultad de Informática, UCM

5. Publicación para el Congreso Español de informática





12. ANEXOS

12.1. GUÍAS DE INSTALACIÓN Y CONFIGURACIÓN

12.1.1. CONSIDERACIONES GENERALES

La distribución final de SmartCloud está formada por dos imágenes en formato de disco virtual de VirtualBox que deben ser ejecutadas con VirtualBox de la manera que se indica más adelante.

Estas imágenes son distribuciones de Ubuntu 10.04 con las herramientas y las bibliotecas necesarias.

El proyecto está diseñado para ser una solución "Out of the box", lo que significa que las imágenes requieren únicamente configuraciones mínimas para funcionar.

La distribución cuenta con dos imágenes:

- **Nodo:** Nodo de cómputo, puede ser instalado en la cantidad de máquinas que hayan disponibles.
- **Server:** Servidor, debe ser instalada solo una vez sobre un servidor de altas prestaciones.

A continuación se detallan las configuraciones mínimas que hay que realizar en ambas imágenes.

12.1.2. VIRTUALIZACIÓN DEL NODO

Como hemos visto antes, la infraestructura del nodo está compuesta fundamentalmente por una máquina virtual que corre sobre Virtual Box. Para cada nodo que se desee instalar en una máquina se procede de igual forma.






A continuación se describen los pasos a seguir para virtualizar cada nodo de forma genérica. La configuración posterior de cada nodo para ejecutar la funcionalidad de nodo de cálculo se especifica en apartados posteriores.

Para configurar el nodo seguimos los siguientes pasos:

1. Arranque VirtualBox, mostrará una vista similar:



Ilustración 12 VirtualBox

2. Seleccione  para crear una nueva máquina virtual y establezca el nombre y especifique el sistema operativo, y haga click en Siguiente. Puede llamar a la máquina cómo desee, en nuestro caso se llamará Nodo.

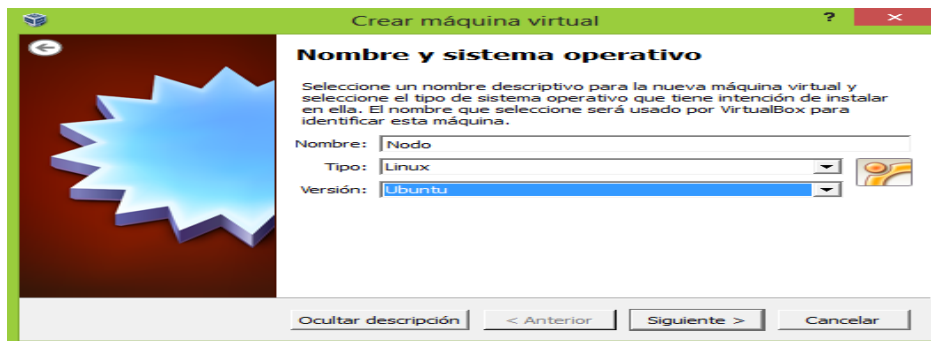


Ilustración 13 Nueva Máquina Virtual

3. Establezca el tamaño de la memoria RAM y haga click en Siguiente.

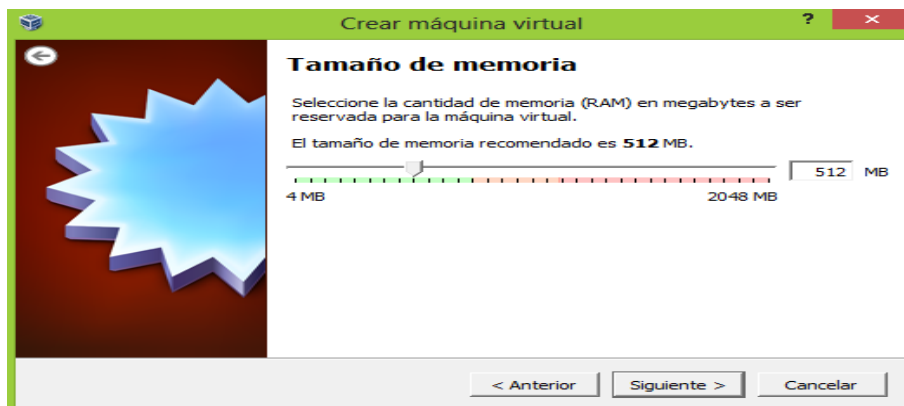



Ilustración 14 Establecer Tamaño de RAM





4. El siguiente paso es establecer la unidad de disco duro. Elegimos la opción "Usar un archivo de disco duro virtual existente" y con el navegador de fichero del botón  seleccionamos el ficho .iso correspondiente a nuestro nodo y hacemos click en Crear.

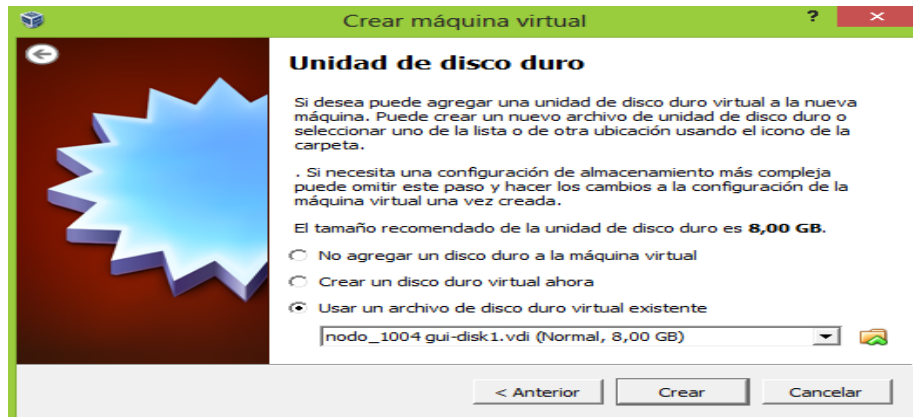


Ilustración 15 Imagen del Nodo

Dependiendo del sistema operativo host, hay que tener en cuenta ciertas limitaciones. En los host Linux, la funcionalidad es limitada cuando se utilizan las interfaces inalámbricas para redes en puente. Actualmente, Virtual Box sólo soporta IPv4; para otros protocolos como IPv6 e IPX debe elegir una interfaz de cable.

Virtual Box nos permite escoger entre los siguientes modos de conexión:

- *No conectado:* Virtual Box muestra un adaptador de red pero sin conexión (cable desconectado).
- *"Network Address Translation" (NAT):* Permite funcionalidad básica desde el sistema operativo Huésped. Navegar por internet acceder al correo, descargar ficheros. Tiene bastantes limitaciones si tenemos que establecer conexiones con la máquina virtual.
- *Adaptador puente:* Simula una conexión física real a la red, asignando una IP al sistema operativo huésped. Esta IP se puede obtener por DHCP o directamente configurándola en el Sistema Operativo huésped.
- *Red interna:* Similar al Adaptador puente, se puede comunicar directamente con el mundo exterior con la salvedad de que ese mundo exterior está restringido a las maquinas virtuales conectadas en la misma red interna. Esta limitación viene justificada por seguridad y velocidad.
- *Adaptador sólo-anfitrión:* Es una mezcla entre los tipos "Adaptador puente" e "interna".

Por defecto tendremos configurado el modo NAT. Para nuestro caso esta configuración no es la más adecuada, ya que no tenemos "visibilidad" del sistema






operativo huésped desde el sistema operativo anfitrión, y nos resulta muy útil tener una IP diferente para cada uno.

Queremos que nuestra máquina virtual se conecte con el servidor a través de la red, para ello establecemos una configuración Bridged Networking (redes en puente). Al activarla, VirtualBox se conecta a una de sus tarjetas de red instaladas e intercambia los paquetes directamente, evitando la pila del sistema operativo del host.

Con un puente de red, VirtualBox utiliza un controlador de dispositivo en el sistema servidor que filtra los datos de su adaptador de red físico. Este controlador se denomina por tanto un "filtro de red" del conductor. Esto permite que VirtualBox para interceptar los datos de la red física e inyectar datos en ella, creando una nueva interfaz de red en el software. Cuando un huésped está utilizando como una interfaz de software nuevo, se ve que el sistema anfitrión como si el huésped se conecta físicamente a la interfaz mediante un cable de red: el host puede enviar datos al huésped a través de esa interfaz y recibir datos de él. Esto significa que se puede configurar enrutamiento o puente entre el cliente y el resto de la red.

Para habilitar un puente de red debemos seleccionar la interfaz deseada host de la lista en la parte inferior de la página, que contiene las interfaces de red físicas de los sistemas.

Pasos:

- 1- En la ventana de configuración seleccionamos la pestaña Red.
- 2- En Adaptador 1, marcamos el check **Habilitar adaptador de red**
- 3- En el campo nombre seleccionamos el interfaz que vamos a utilizar: Ethernet, AirPort, etc. Es importante establecer que este dispositivo debe estar conectado para el correcto funcionamiento de la red entre el sistema operativo anfitrión y el huésped. Si elegimos Ethernet y no nos conectamos a una red mediante el cable no se podrán establecer conexiones entre los sistemas operativos.
- 4- Mostramos las configuraciones avanzadas haciendo click en
- 5- Se habilita el modo promiscuo
- 6- En esta ventana también tendremos que definir una dirección MAC propia para el adaptador virtual. Por defecto nos asigna una aleatoria que podremos refrescar mediante el botón que tiene a la derecha. 





SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

7- VirtualBox permite definir entre seis variantes de hardware virtualizado.

- AMD PCNet PCI II (Am79C970A)
- AMD PCNet FAST III (Am79C973)
- Intel PRO/1000 MT Desktop (82540EM)
- Intel PRO/1000 T Server (82543GC)
- Intel PRO/1000 MT Server (82545EM)
- Paravirtualized network adapter (virtio-net)

Por defecto tendremos seleccionada la 2ª opción ya que es soportada por la mayoría de sistemas operativos. Podemos dejarla tal cual o utilizar otra.

8- Establecida la configuración del adaptador, hacemos click en Aceptar.

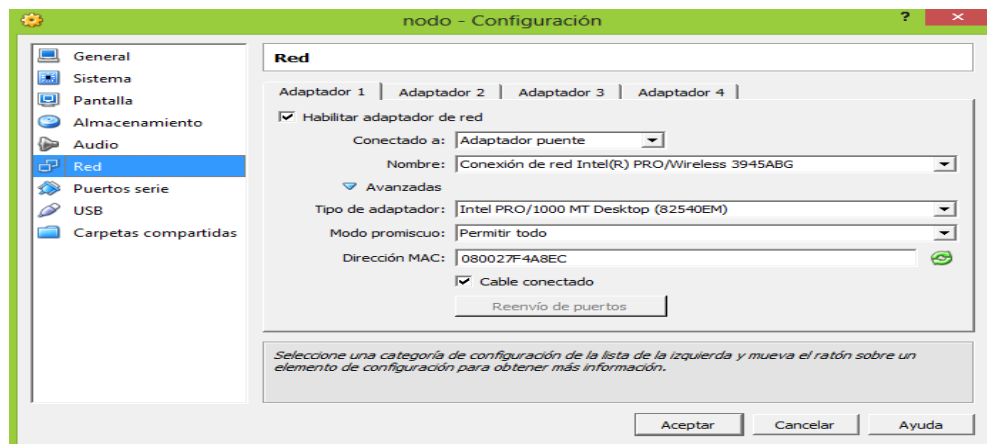


Ilustración 16 Configuración de la Red

9- Comprobamos que la configuración es correcta y el nodo se conecta correctamente a la red.

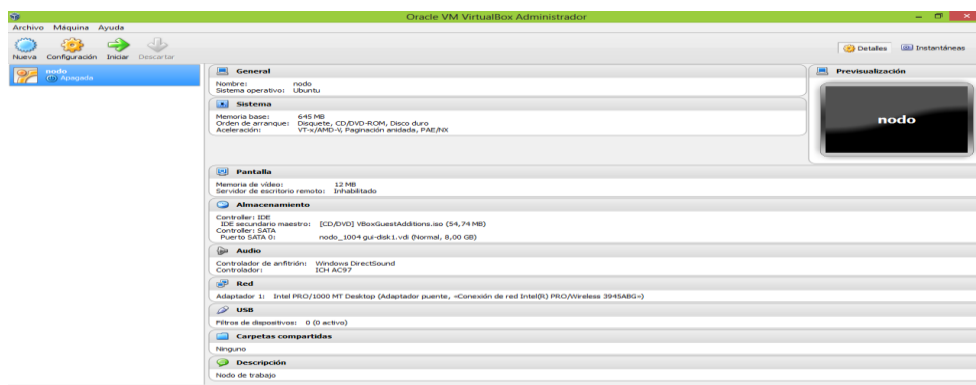


Ilustración 17 Nodo de Cálculo configurado

12.1.2.1. ESTABLECER CONTROL DE ARRANQUE Y APAGADO DEL NODO



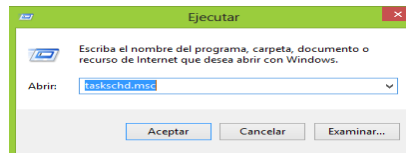


SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

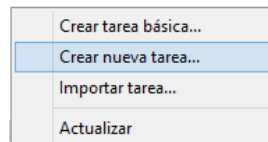
Tanto el servidor como el host donde está instalado el VirtualBox deben saber si la instancia de la máquina virtual está disponible para trabajar como nodo. Para ello se añaden dos tareas programadas que controlan el arranque y apagado de los nodos y notifican al servidor sus estados.

Para establecer las tareas se siguen los siguientes pasos:

- 1- Abrimos el Programador de tareas: ejecutando taskschd.msc

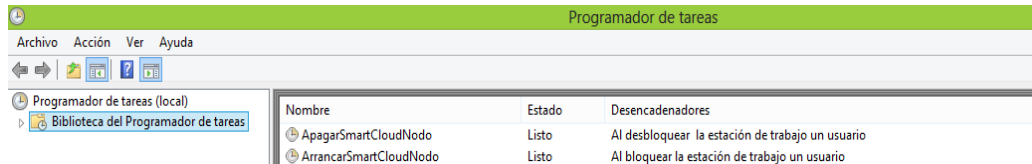


- 2- Hacemos click derecho sobre la biblioteca de tareas y añadimos una nueva tarea



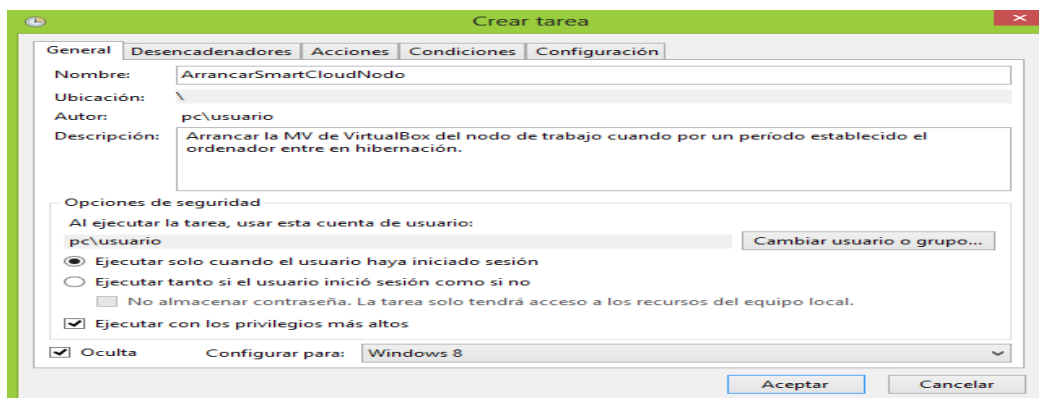
- 3- Se añade la tarea correspondiente.

Las tareas quedan programadas y se ejecutarán cuando los desencadenadores establecidos se den.



12.1.2.2. AÑADIR TAREA ARRANCAR MÁQUINA VIRTUAL

1. En el panel General: asignamos un nombre a la tarea y establecemos los parámetros de configuración para el sistema operativo.





SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

2. En el panel Desencadenadores: creamos un nuevo desencadenador en el cual establecemos que la tarea debe iniciarse "Al bloquearse la estación de trabajo".

Crear tarea

General | Desencadenadores | Acciones | Condiciones | Configuración

Cuando se crea una tarea, se pueden especificar las condiciones que la activarán.

Desencadenador	Detalles	Estado
Al bloquearse la est...	Al bloquear la estación de trabajo un usuario	Habilitado

Nuevo... | Editar... | Eliminar

Aceptar | Cancelar

Nuevo desencadenador

Iniciar la tarea: Al bloquearse la estación de trabajo

Configuración

Cualquier usuario
 Usuario específico: pplusuario Cambiar usuario...

Configuración avanzada

Retrasar durante: 15 minutos

Repetir cada: 1 hora durante: 1 día

Detener todas las tareas en ejecución al final de la duración de repetición

Detener la tarea si se ejecuta durante más de: 3 días

Activación: 18/02/2013 0:38:37 Sincronizar zonas horarias

Expiración: 18/02/2014 0:38:37 Sincronizar zonas horarias

Habilitado

Aceptar | Cancelar

3. En el panel Acciones: establecemos la acción a realizar. En este caso al añadir la nueva acción se define que se ejecutará un programa y se busca el fichero "TareaArrancarMVNode". Este programa inicia el control del nodo y comunica al servidor que ya está disponible para realizar una tarea.

Crear tarea

General | Desencadenadores | Acciones | Condiciones | Configuración

Al crear una tarea, debe especificar la acción que se producirá cuando se inicie la tarea.

Acción	Detalles
Iniciar un programa	"C:\Users\usuario\SMARTCLOUD\nodo_ordenador del funcionario\Script

Nueva... | Editar... | Eliminar

Aceptar | Cancelar

Editar acción

Especifique la acción que debe realizar esta tarea.

Acción: Iniciar un programa

Configuración

Programa o script:
"\\TareaArrancarMV.cmd" Examinar...

Agregar argumentos (opcional):

Iniciar en (opcional):

Aceptar | Cancelar

4. En el Panel Condiciones: establecemos las condiciones de ejecución.





SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

Crear tarea

General | Desencadenadores | Acciones | Condiciones | Configuración

Especifique las condiciones que, junto a los desencadenadores, determinarán si se debe ejecutar la tarea. Si alguna de las condiciones especificadas no se cumple, no se ejecutará la tarea.

Inactivo

Iniciar la tarea solo si el equipo está inactivo durante: 10 minutos

Esperar a que esté inactivo durante: 1 hora

Detener si el equipo deja de estar inactivo

Reiniciar si el estado de inactividad se reanuda

Energía

Iniciar la tarea solo si el equipo está conectado a la corriente alterna

Detener si el equipo empieza a usar la batería

Activar el equipo para ejecutar esta tarea

Red

Iniciar solo si la siguiente conexión de red está disponible: Cualquier conexión

Aceptar Cancelar

5. En el panel Configuración: establecemos el funcionamiento de la tarea.

Crear tarea

General | Desencadenadores | Acciones | Condiciones | Configuración

Especifique la configuración adicional que afecta al comportamiento de la tarea.

Permitir que la tarea se ejecute a petición

Ejecutar tarea lo antes posible si no hubo inicio programado

Si la tarea no se ejecuta, reiniciarla cada: 1 minuto

Intentar el reinicio un máximo de: 3 veces

Detener la tarea si se ejecuta durante más de: 3 días

Detener tarea en ejecución si no finaliza cuando se solicite

Eliminar tareas no reprogramadas después de: 30 días

Aplicar la siguiente regla si la tarea ya está en ejecución: No iniciar una instancia nueva

Aceptar Cancelar

12.1.2.3. AÑADIR TAREA APAGAR MÁQUINA VIRTUAL

3- En el panel General: asignamos un nombre a la tarea y establecemos los parámetros de configuración para el sistema operativo.

Crear tarea

General | Desencadenadores | Acciones | Condiciones | Configuración

Nombre: ApagarSmartCloudNodo

Ubicación: \\

Autor: pc\usuario

Descripción: Arrancar la MV de VirtualBox del nodo de trabajo cuando por un período establecido el ordenador se desbloquee.

Opciones de seguridad

Al ejecutar la tarea, usar esta cuenta de usuario: pc\usuario [Cambiar usuario o grupo...]

Ejecutar solo cuando el usuario haya iniciado sesión

Ejecutar tanto si el usuario inició sesión como si no

No almacenar contraseña. La tarea solo tendrá acceso a los recursos del equipo local.

Ejecutar con los privilegios más altos

Ocultar Configurar para: Windows 8

Aceptar Cancelar

4- En el panel Desencadenadores: creamos un nuevo desencadenador en el cual establecemos que la tarea debe iniciarse "Al desbloquearse la estación de trabajo".





SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

Desencadenador	Detalles	Estado
Al desbloquearse la...	Al desbloquear la estación de trabajo un usuario	Habilitado

- 5- En el panel Acciones: establecemos la acción a realizar. En este caso al añadir la nueva acción se define que se ejecutará un programa y se busca el fichero "TareaApagarMVNode". Este programa inicia el control del nodo y comunica al servidor que ya no está disponible para realizar una tarea.

Acción	Detalles
Iniciar un programa	"C:\Users\usuario\SMARTCLOUD\nodo_ordenador del funcionario\Script"

- 6- En el Panel Condiciones: establecemos las condiciones de ejecución.

Inactivo

Iniciar la tarea solo si el equipo está inactivo durante: 10 minutos

Esperar a que esté inactivo durante: 1 hora

Detener si el equipo deja de estar inactivo

Reiniciar si el estado de inactividad se reanuda

Energía

Iniciar la tarea solo si el equipo está conectado a la corriente alterna

Detener si el equipo empieza a usar la batería

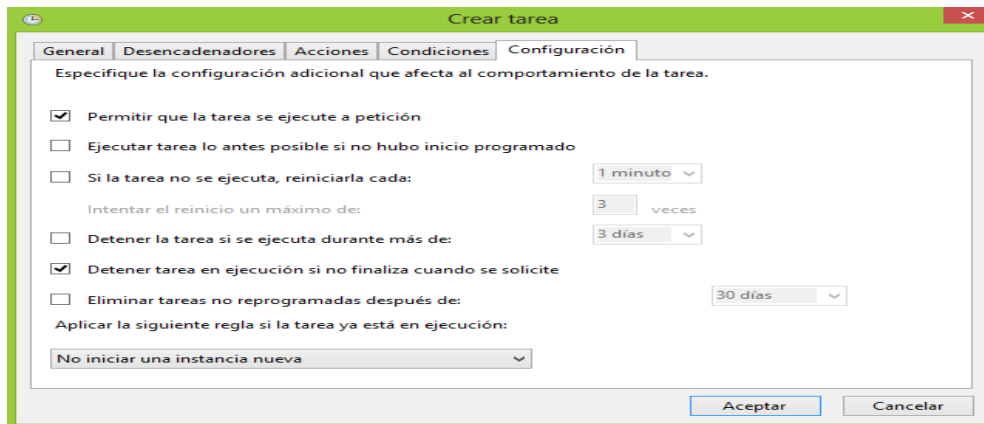
Activar el equipo para ejecutar esta tarea

Red

Iniciar solo si la siguiente conexión de red está disponible: Cualquier conexión

- 7- En el panel Configuración: establecemos el funcionamiento de la tarea.





12.1.3. CONFIGURACIÓN

Para el proceso de configuración del servidor y de los nodos, hemos decidido una arquitectura sencilla formada por el servidor y dos nodos.

Partimos de la premisa que el servidor ha sido instalado y los nodos virtualizados siguiendo los pasos especificados en el apartado anterior.

12.1.3.1. DENTRO DE LA IMAGEN SERVER

12.1.3.1.1. CONFIGURAR LOS NODOS PARA SER ACCESIBLES POR EL SERVIDOR

Ejecutar:

```
server@server:~$ sudo gedit /etc/hosts
```

El archivo debe contener lo siguiente:

```
127.0.0.1 localhost
127.0.1.1 server
192.168.56.102 nodo1
192.168.2.6 nodo2
```

*Nota: en este caso los nodos existentes en el sistema son nodo1 con la ip 192.168.56.102 y nodo2 con la ip 192.168.2.6. Esto depende de cada infraestructura.

Al añadir nodos al clúster, deben añadirse a este fichero. El nombre del nodo debe ser el que se encuentra dentro del fichero /etc/hostname del nodo.

Para probar esta configuración basta con ejecutar `ping nodoN`

12.1.3.1.2. CONFIGURAR LOS NODOS PARA SER ACCESIBLES POR PBS_SERVER

Ejecutar:

```
server@server:~$ sudo gedit /var/spool/torque/server_priv/nodes
```





El archivo debe contener lo siguiente:

```
nodo1 np=4 smartcloud  
nodo2 np=2 smartcloud
```

*Nota: en este caso los nodos existentes en el sistema son nodo1 y nodo2. Esto depende de cada infraestructura. Se puede especificar el número de procesadores de cada nodo. Pbs_server asignará un trabajo por procesador disponible.

Es fundamental que en cada línea del archivo se encuentre la frase "smartcloud" como se muestra en el ejemplo, esto indica que el gestor pbs los agrupa dentro un grupo de recurso.

Al añadir nodos al clúster, deben añadirse a este fichero. El nombre del nodo debe ser el que se encuentra dentro del fichero /etc/hostname del nodo

Para probar esta configuración basta con reiniciar el pbs_server y ejecutar pbsnodes

12.1.3.1.3. SCRIPTS INICIALES

Al iniciar el servidor se deben ejecutar los siguientes scripts en orden. (Se pueden automatizar en el directorio init.d)

1)nfs.sh:

```
#!/bin/bash  
echo "Starting NFS"  
sudo service portmap start  
sudo service nfs-kernel-server start  
echo "NFS started"  
echo "shared folder: /home/server/shared"
```

2)Pbs.sh:

```
#!/bin/bash  
echo "Starting pbs_server"  
sudo trqauthd start  
sudo pbs_server  
echo "Starting scheduler"  
sudo pbs_sched
```

12.1.3.2. DENTRO DE LA IMAGEN NODO

12.1.3.2.1. CONFIGURAR SERVIDOR DENTRO DE LOS NODOS

Ejecutar:





```
server@server:~$ sudo gedit /etc/hosts
```

El archivo debe contener lo siguiente: (en este ejemplo nos encontramos en el nodo1)

```
127.0.0.1 localhost
192.168.56.103server
```

*Nota: en este caso el servidor con nombre `server` tiene la ip 192.168.36.103. Esto depende de cada infraestructura. Durante la instalación de este nodo debe modificarse este fichero con la ip del servidor.

Para probar esta configuración basta con ejecutar `ping server`

12.1.3.2.2. CONFIGURAR EL NOMBRE DE CADA NODO

*Nota: El nombre de cada nodo debe ser único.

Editar el archivo `/etc/hostname` ejecutando:
`server@server:~$ sudo gedit /etc/hostname`

```
nodo1
```

*Nota: En el ejemplo nos encontramos en el nodo1, pero hay que cambiarlo con cada nodo nuevo. Los nodos deben tener un nombre único en toda la infraestructura.

12.1.3.2.3. SCRIPTS INICIALES

Al iniciar el nodo se deben ejecutar los siguientes scripts en orden. (Se pueden automatizar en el directorio `init.d`)

1)nfs_client.sh:

```
#!/bin/bash
echo "Starting NFS"
sudo service portmap start
sudo mount -v server:/home/server/shared /home/server/shared
echo "NFS started"
echo "shared folder: /home/server/shared"
```

2)pbs_mom.sh:

```
#!/bin/bash
echo "Starting pbs_mom"
sudo pbs_mom -D
echo "pbs_mom started"
```



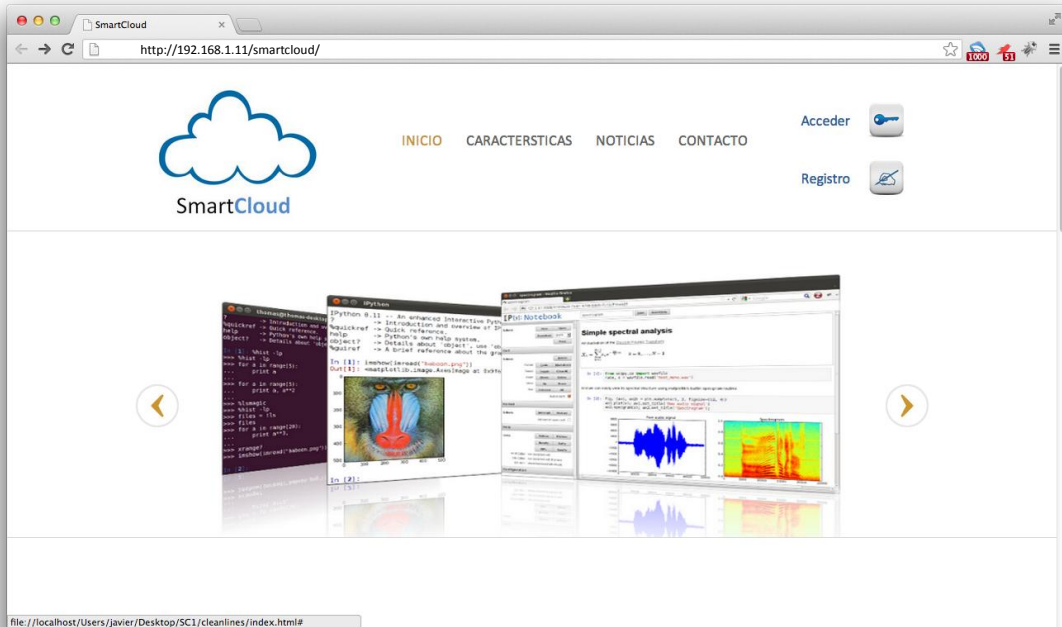


12.2. MANUAL DE USUARIO

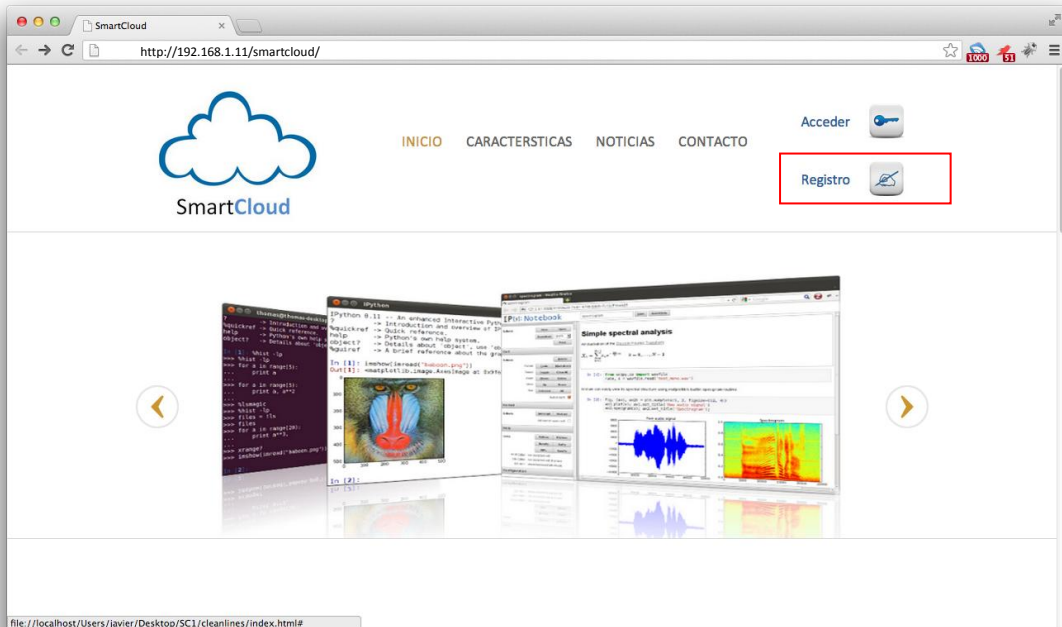
12.2.1. FRONTEND

12.2.1.1. REGISTRO EN SMARTCLOUD.

El usuario accede a <http://192.168.1.11/smartcloud/>



1. Presionar el botón "Registro" en la interfaz web





SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

2. Hacer click en el botón "Registro"
3. Introducir datos personales y presiona "Aceptar"
 - Escribir un nombre de usuario y contraseña, esta información la necesitara para volver acceder sin tener que volver a registrarse.
 - Seleccionar si desea utilizar autenticación con Google Drive (opcional)

SmartCloud

INICIO CARACTERSTICAS NOTICIAS CONTACTO

Acceder

Registro

Username: prueba

Password: ●●●●●●

Vincular con Google Drive? Si No

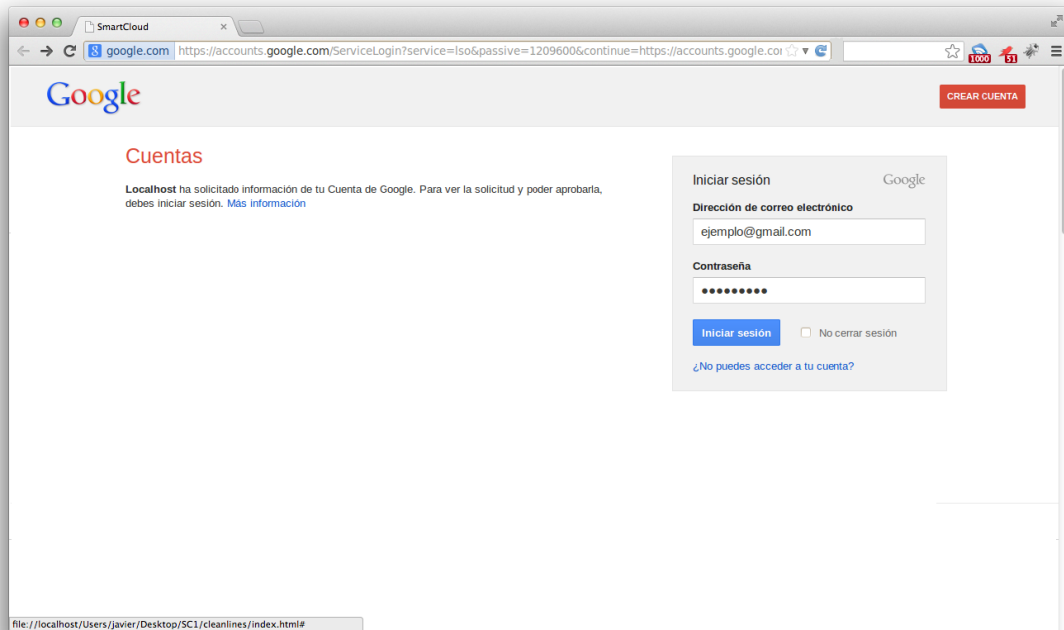
Aceptar Cancelar



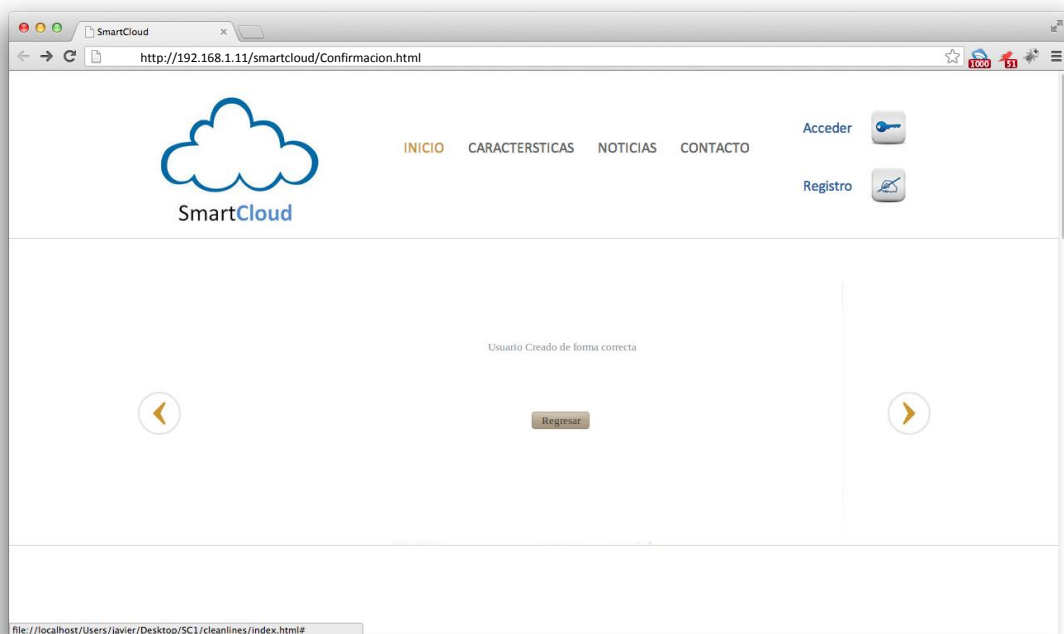


SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

5. Luego de aceptar la vinculación con Google Drive (pulsando el botón Si), tiene que rellenar sus datos de su cuenta de Google para poder asociarla.



6. El sistema notifica al usuario de la creación del usuario

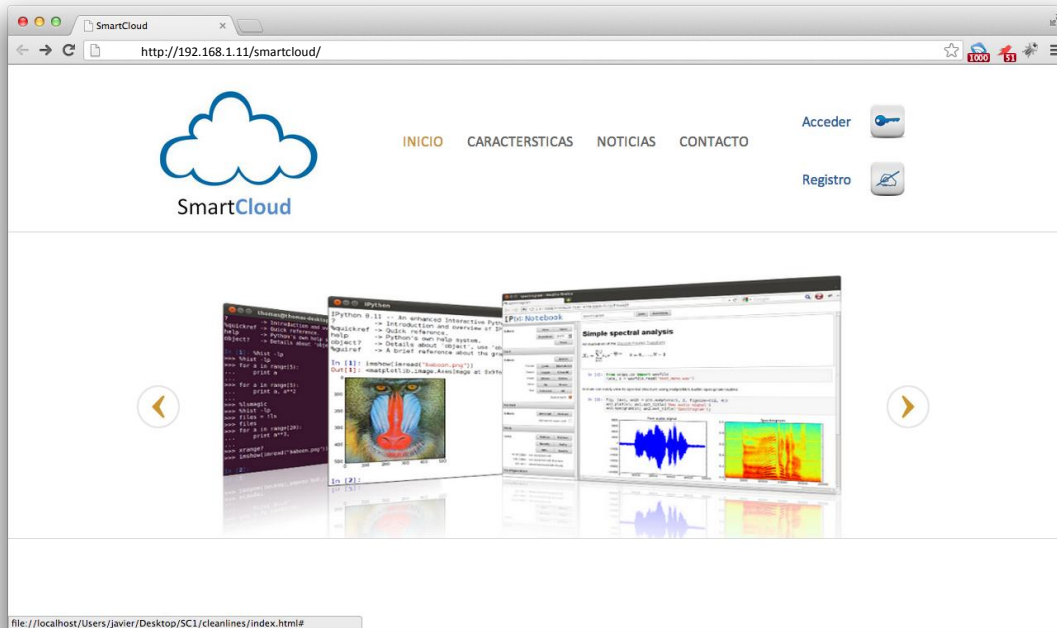




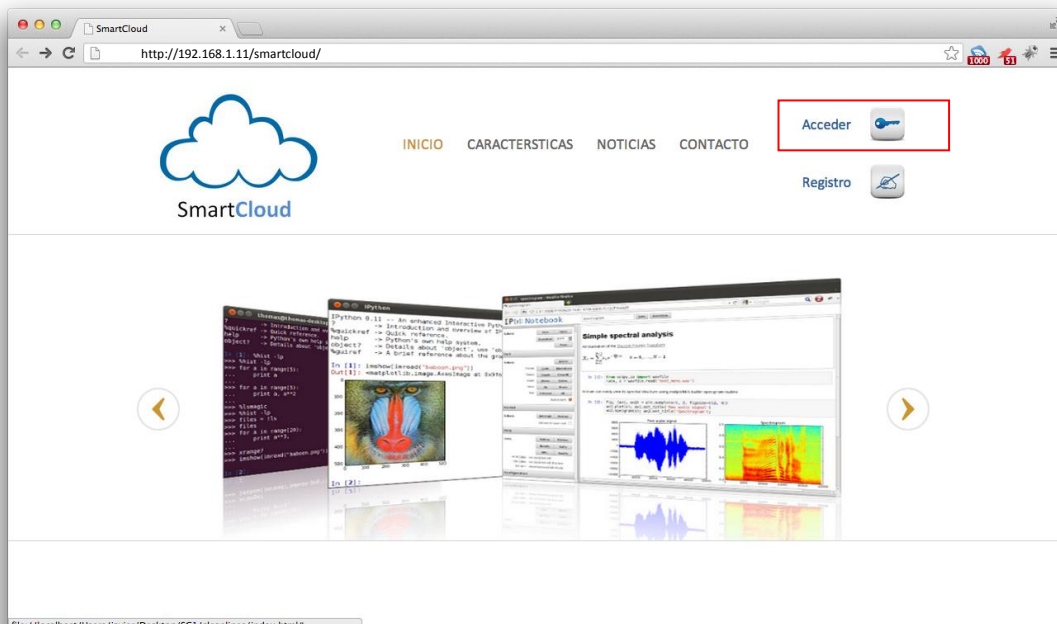
SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

12.2.1.2. LOGIN CON GOOGLE ID

1. El usuario accede a la aplicación <http://192.168.1.11/smartcloud/>

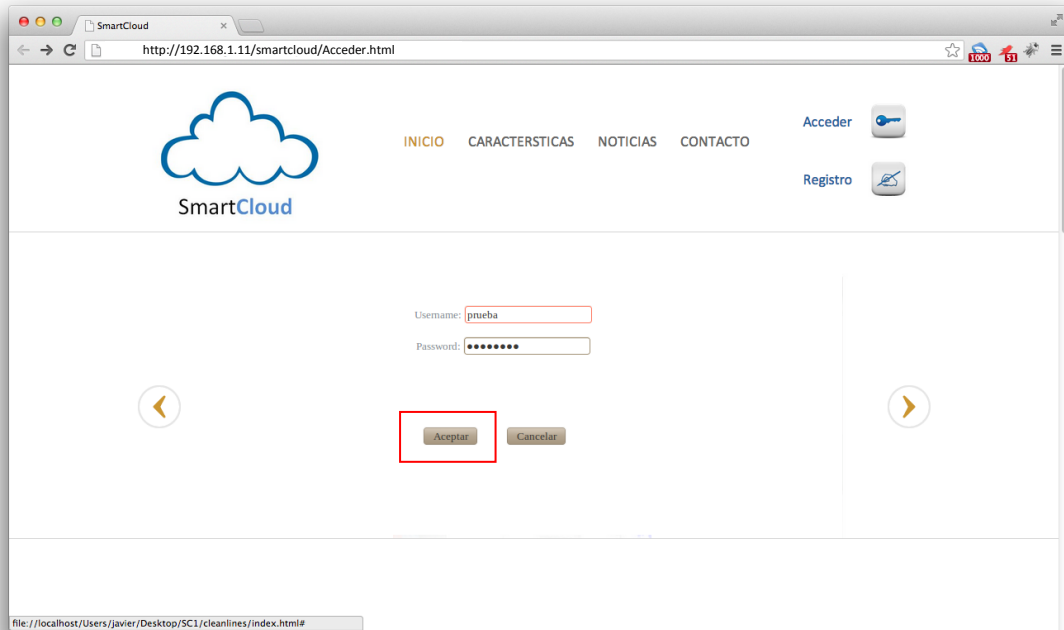


2. Presionar el botón "Acceder" en la interfaz web



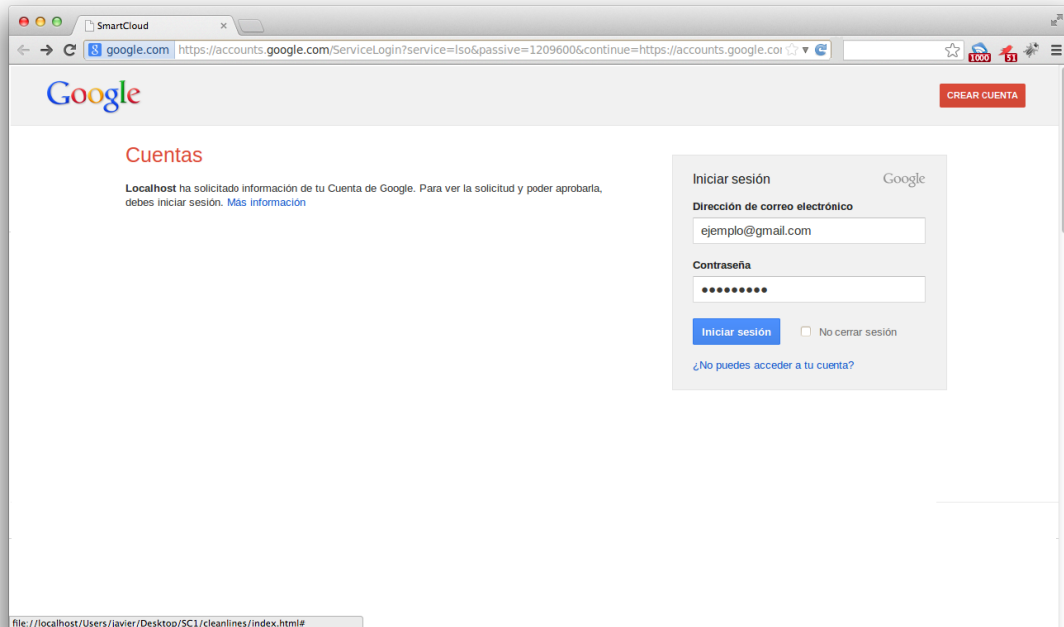


3. Login de SmartCloud



4. Login de Google

- Luego de ser redirigido tiene que rellenar sus datos de su cuenta de Google para poder acceder a su drive, después pulsar sobre iniciar sesión.



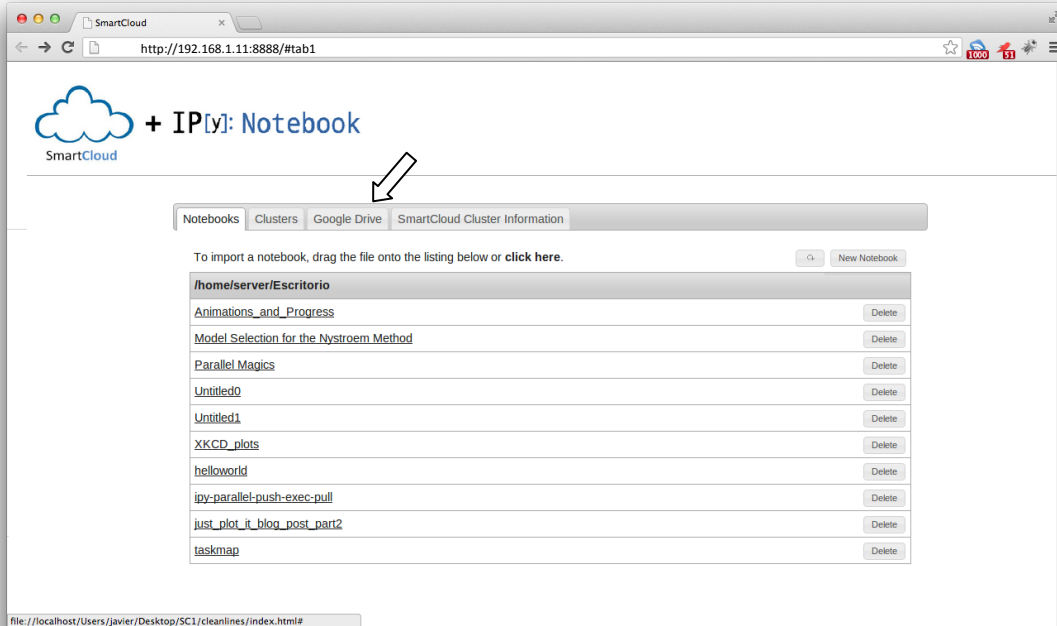
5. Es redirigido a la interfaz de iPython con sus archivos personales y la pestaña de "Google Drive" activada.





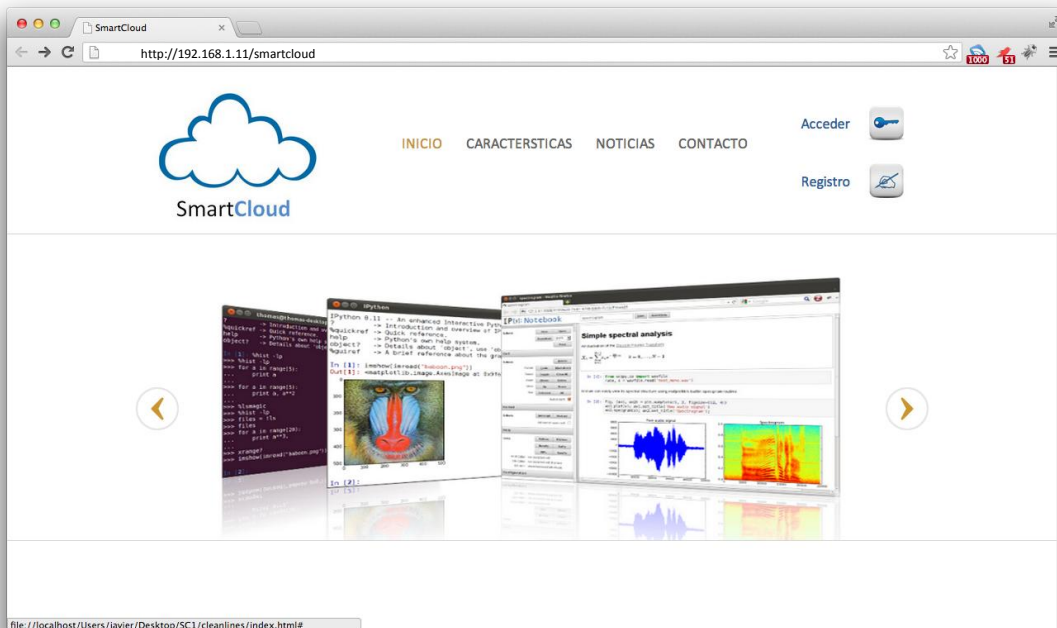
SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

- Si la autenticación de Google se ha resuelto satisfactoriamente el usuario podrá acceder al notebook de iPython. En caso contrario se le volverá a pedir el usuario o contraseña.



12.2.1.3. LOGIN SIN GOOGLE ID

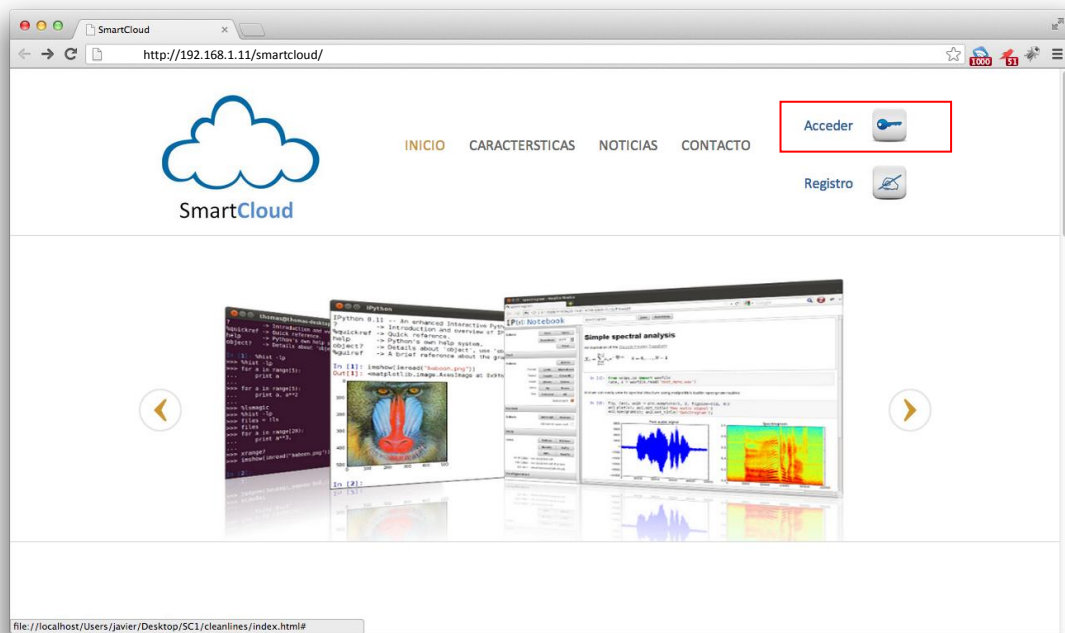
1. Acceder a la aplicación `http://192.168.1.11/smartcloud/`





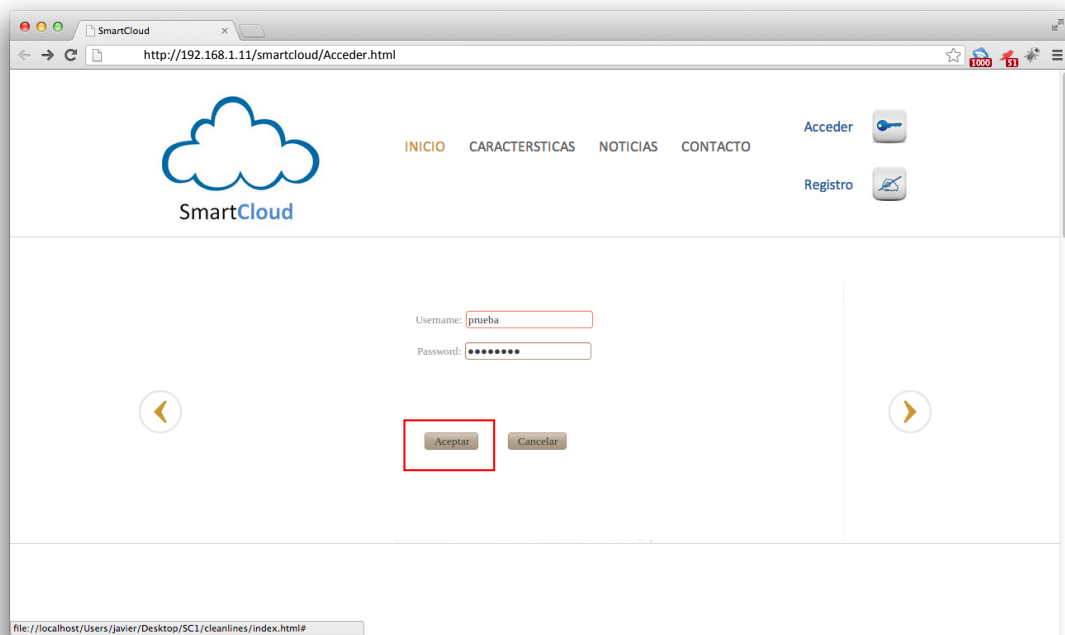
SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

2. Presionar el botón "Acceder" en la interfaz web



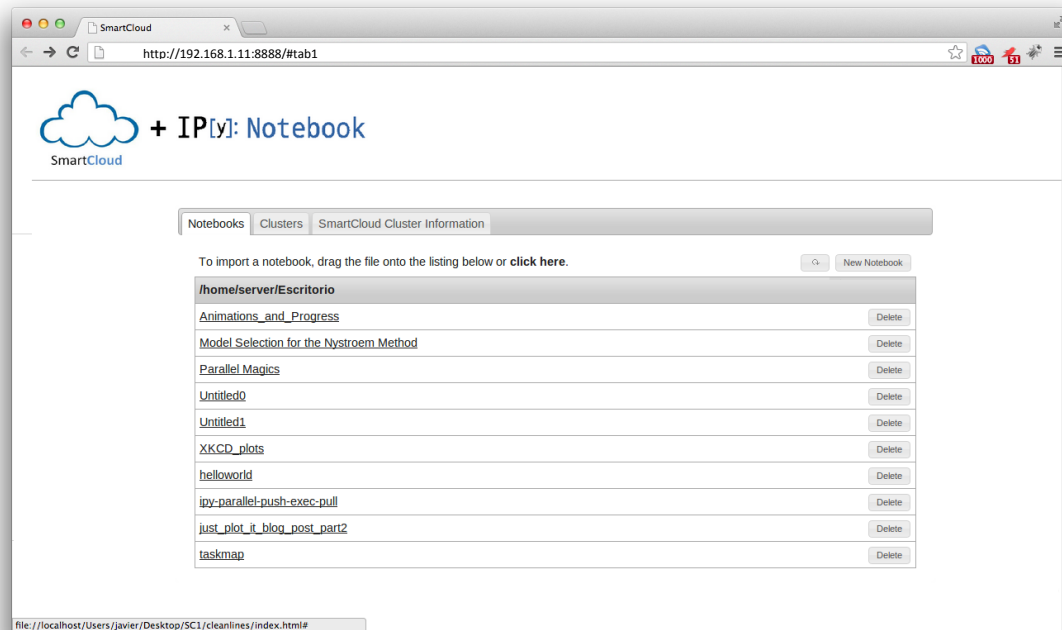
3. Es redirigido al login propio de SmartCloud

- Una vez rellenados el nombre de usuario y contraseña de forma correcta pulsa sobre el botón aceptar.





4. Se accede a la interfaz de IPython con sus archivos personales.

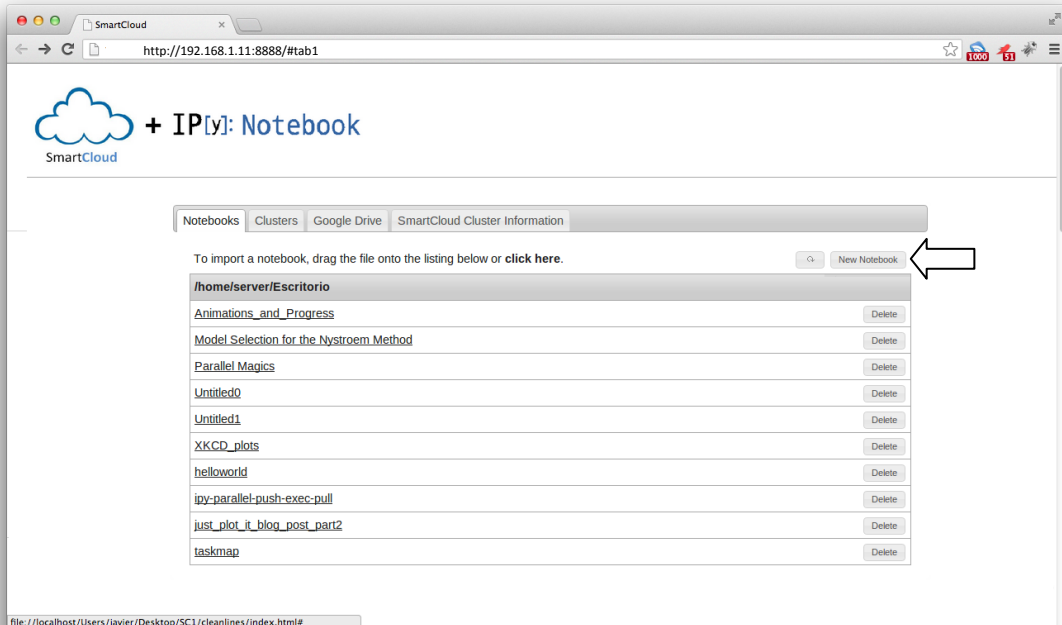




12.2.2. NOTEBOOK

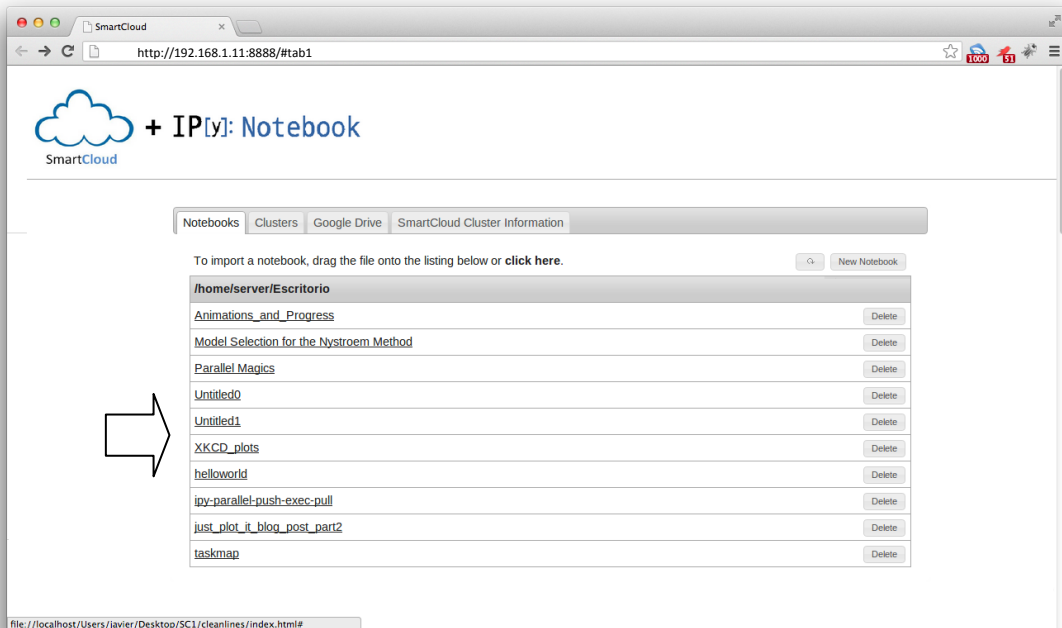
12.2.2.1. CREAR UN NOTEBOOK

Para crear un notebook, se pulsa sobre el botón de "new notebook".



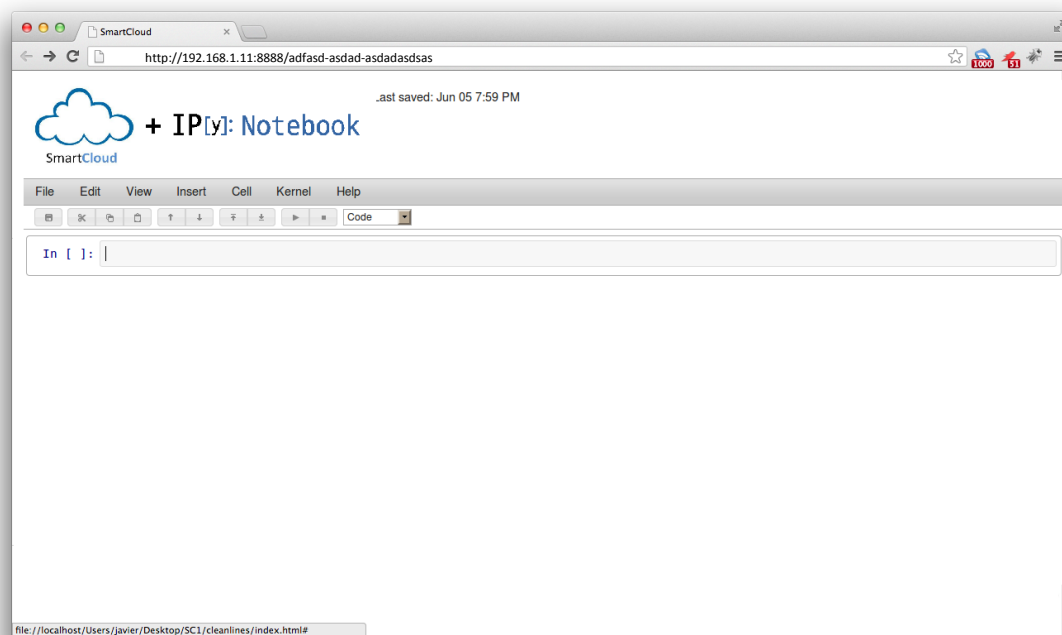
12.2.2.2. ABRIR UN NOTEBOOK

1. Para abrir un notebook se hace click sobre alguno de la lista que le aparece.





2. Después de seleccionar el notebook, le aparece la siguiente imagen



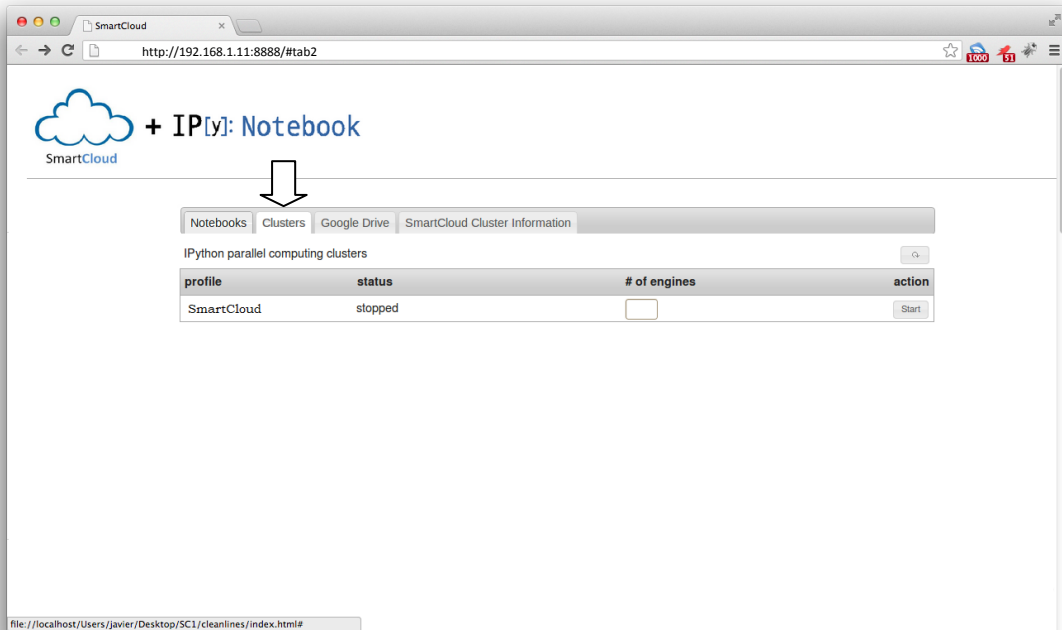
Para más información sobre el uso del notebook de iPython consulte el manual en la página oficial.



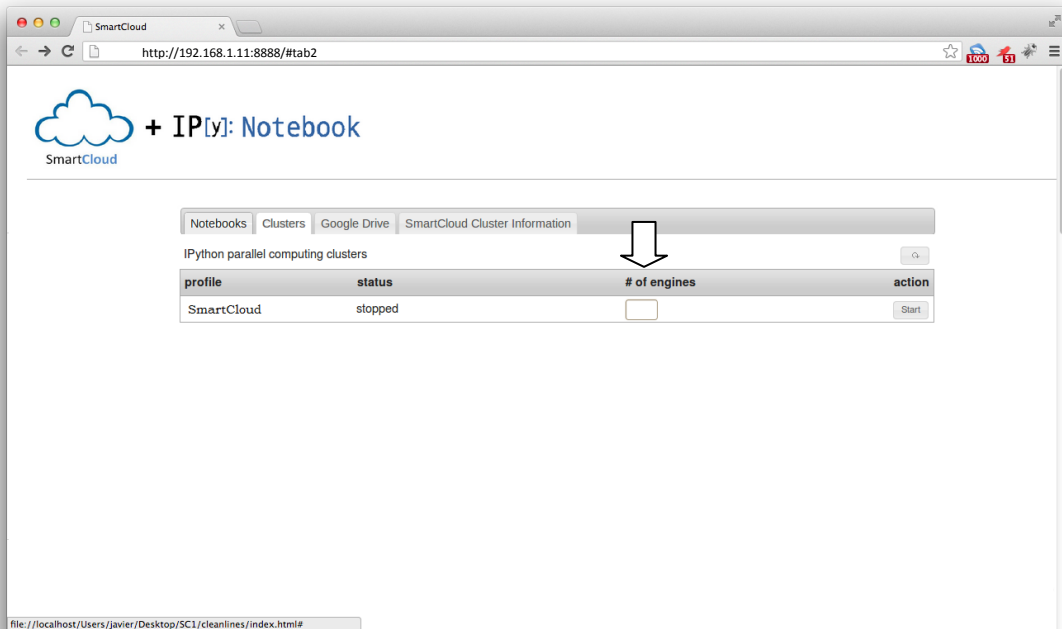


12.2.2.3. CONFIGURAR RECURSOS DEL CLÚSTER

1. Acceder a la pestaña "Cluster"

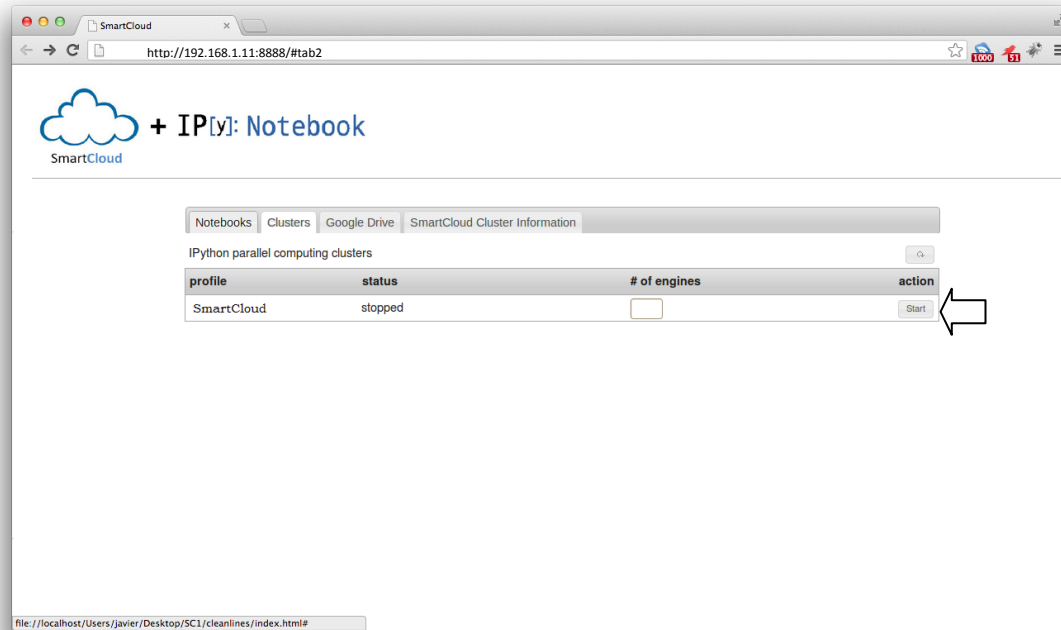


2. Ingresar el número de procesadores en el clúster



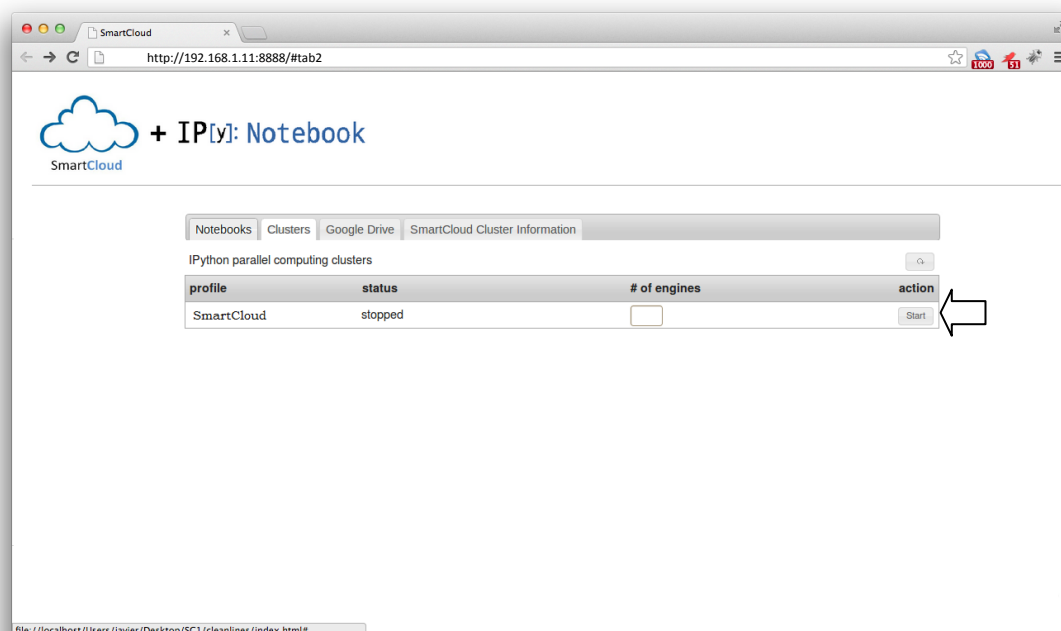


3. Presionar 'Start'



12.2.2.4. DETENER EL CLÚSTER

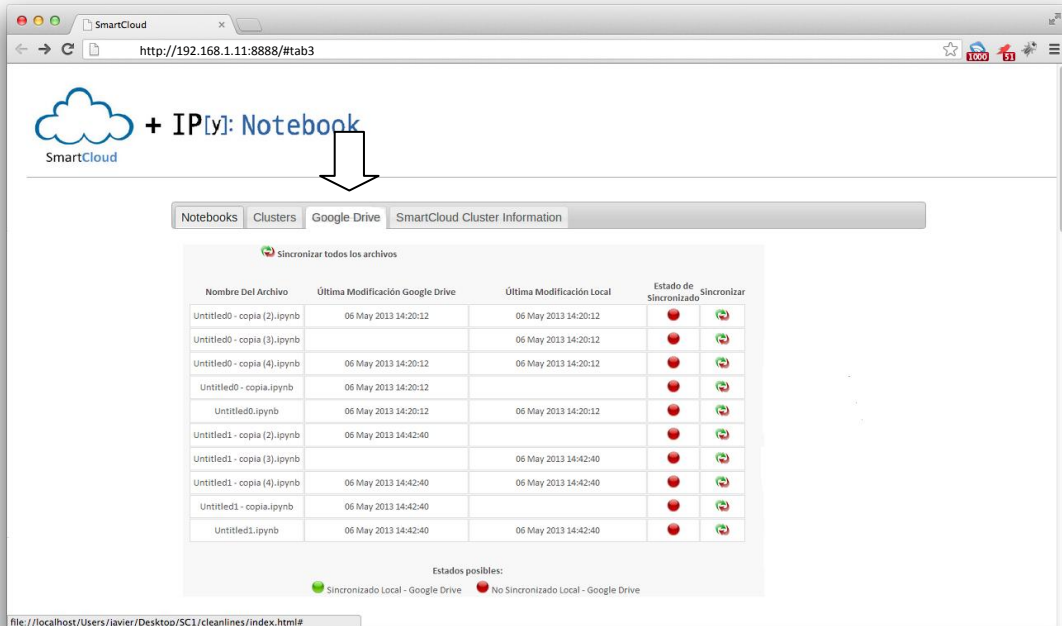
Para poder detener el clúster tiene que pulsar sobre Stop.



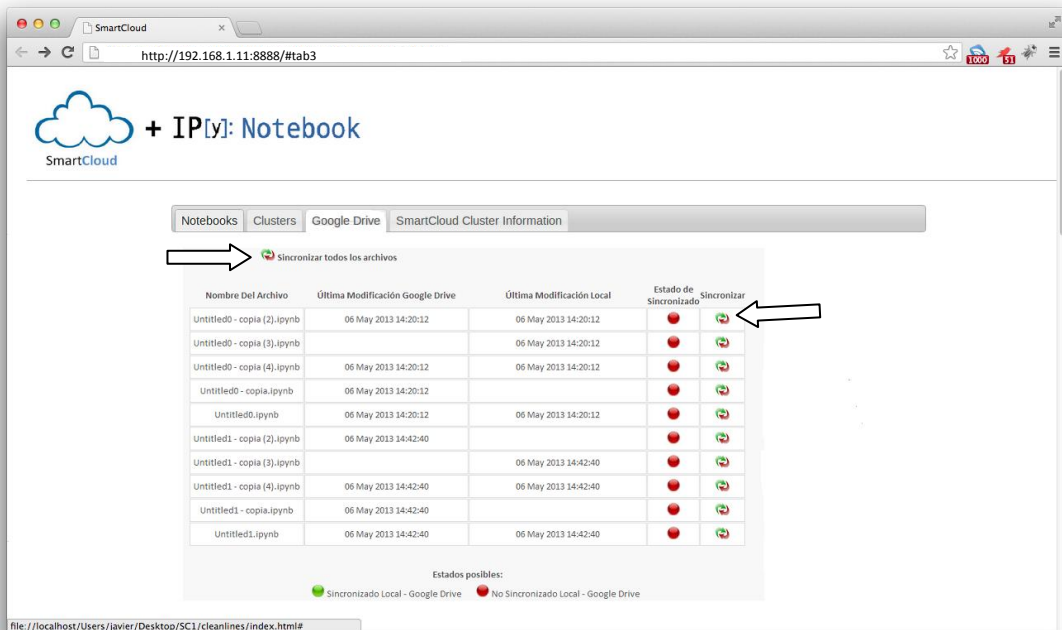


12.2.2.5. SINCRONIZAR ARCHIVOS EN GOOGLE DRIVE

1. Acceder a la pestaña Google drive

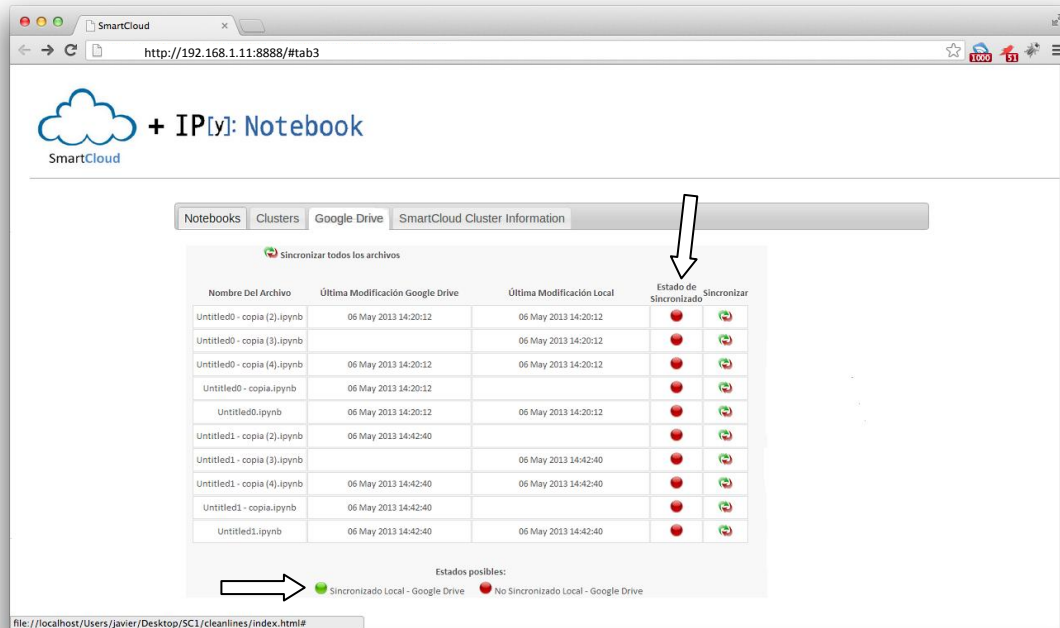


2. El usuario presiona 'Sincronizar todos' o 'Sincronizar un archivo específico'
- Para sincronizar los archivos tenemos 2 mecanismos, para sincronizarlo individualmente o sino todos de forma simultanea



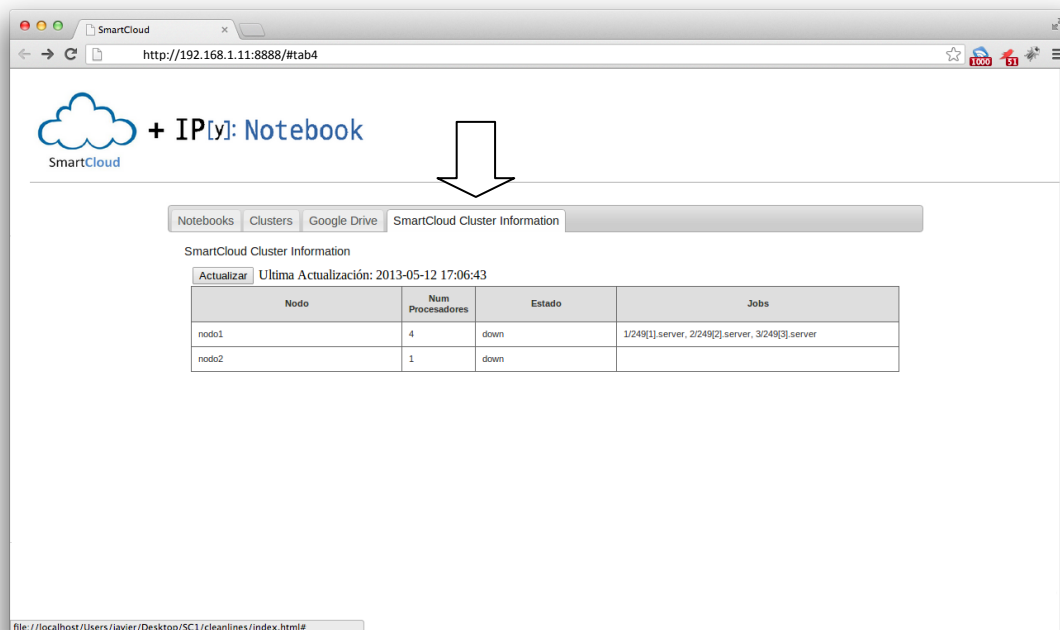


3. Para saber si un archivo esta sincronizado o no solo tenemos que ver que su estado de sincronización tenga un circulo verde



12.2.2.6. CONSULTAR EL ESTADO DEL CLÚSTER

1. Acceder a la pestaña SmartCloud clúster información





2. Presionar 'Actualizar'

The screenshot shows a web browser window with the URL `http://192.168.1.11:8888/#tab4`. The page title is "SmartCloud + IP[y]: Notebook". There are navigation tabs for "Notebooks", "Clusters", "Google Drive", and "SmartCloud Cluster Information". The "SmartCloud Cluster Information" tab is active, showing a table of cluster data. A white arrow points to the "Actualizar" button. The table shows two nodes, both with a "down" status.

Nodo	Num Procesadores	Estado	Jobs
nodo1	4	down	1/249[1_server, 2/249[2_server, 3/249[3_server
nodo2	1	down	

3. El sistema actualiza la información del clúster





12.3. ARTÍCULOS DE REFERENCIA Y PUBLICACIONES

12.3.1. ARTÍCULO EN HPC IN THE CLOUD

VISUAL ANALYTICS
See your data for all it's worth.
[LIVE DEMO: TRY IT NOW](#)



HPC Job Bank
HPC EMPLOYERS
Posting listings online is easy!
[Register Now](#)

HPC In the Cloud

Dedicated to covering high-end cloud computing in science, industry and the datacenter

Seleccionar idioma

Subscribe | Sign In



Search

Home News Topics Sectors Market Watch Events Resources Job Bank About



January 04, 2013 Student Projects Highlight Cloud's Potential

Jose Luis Vazquez-Poletti, Assistant Professor in Computer Architecture at Complutense University of Madrid

Page: 1 | 2



SmartCloud (2012-2013, ongoing)

This project is focused on getting the most value out of existing infrastructure, as well as providing a service to UCM researchers or any member of an academic environment.



Tabor Communications
Corporate Video

Top News from
Leading HPC in the
Cloud Solution
Providers



SmartCloud Team. César Cayo, Javier Bachrachas and Ailyn Baitá at the Monument to the Spanish Constitution at Madrid.

The main idea is to host virtual machines on computers in a LAN that aren't currently in use. These machines don't need to reside in a computer laboratory. Users from the administrative staff also qualify to be in the SmartCloud resource pool.

These machines could handle several guest types, which would serve as processing nodes inside a computing cluster or even separate machines for interactive access. One of the approaches is to rely on iPython for providing the researchers access to a powerful Matlab-style web notebook, which would be linked to their Google Drive accounts, sharing their data easily.

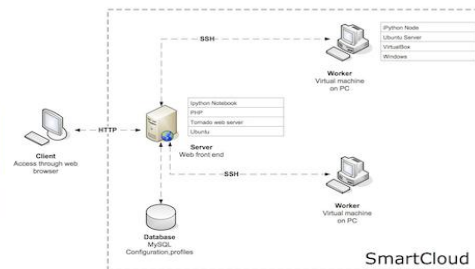


Diagram showing SmartCloud functionalities.

All technologies used in this project are open source and the outcome will represent a direct save for the adopting institution when it comes to licenses and hardware.

Both CygnusCloud and SmartCloud Master Thesis projects are co-advised by my colleague Dr. Jose Antonio Martin H., who is a Python expert and whose research interests are mainly Soft-Computing and Computational Intelligence. Indeed, this collaboration has increased the interdisciplinary nature and the quality of the solutions provided.

This Just In Most Read Behind the Cloud

This Week's Headlines

- Logiworks Launches Managed AWS Service
- Infinitely Virtual Rolls Out Implementation of New VMware vSphere 5.1 Platform
- Spotify and NASA Join Strong Keynote Program at Cloud World Forum
- Huawei Helps Build High-Performance Hyper-V Network Virtualization Gateway

VISUAL ANALYTICS
See your data for all it's worth.
[LIVE DEMO: TRY IT NOW](#)



Short Takes

IBM Acquires SoftLayer

Jun 04, 2013 | IBM hopes to continue being a player in the cloud enterprise business, making a significant stride toward that end today. As announced in a press conference, IBM is looking to expand their cloud portfolio in their acquisition of Dallas-based SoftLayer.

Throwing Cancer on the BonFIRE

May 30, 2013 | Cloud environments featuring more elasticity help along that process by providing hospitals a cost-efficient avenue to run simulations on particular radiation treatments, as highlighted by the presentation below by BonFIRE.

Running Stochastic Models on HTCCondor

May 30, 2013 | In an effort to evaluate creative methods of participating in large scientific projects, research out of Brigham Young University done by Spencer Taylor examined the open source software HTCCondor, which makes use of computing power from idle computers to perform jobs on a local network.

NASA Builds 'Climate in a Box'

May 23, 2013 | The study of climate change is one of those scientific problems where it is almost essential to model the entire Earth to

Newsletters

Stay informed! Subscribe to HPC in the Cloud email Newsletters.

- HPC in the Cloud Update
- HPCwire Weekly Update
- Digital Manufacturing Report
- Datanami
- HPCwire Conferences & Events
- Job Bank
- HPCwire Product Showcases

Enter Email Address

Submit

sas
THE POWER TO KNOW.

VISUAL ANALYTICS
See your data for all it's worth.



[LIVE DEMO: TRY IT NOW](#)





12.3.2. ARTÍCULO EN TRIBUNA COMPLUTENSE

30/01/2013

TRIBUNA COMPLUTENSE

Ciencia

Ignacio Cirac, premio Wolf 2013 y Medalla de Honor del Instituto Niels Bohr



J. DE MIGUEL

Ignacio Cirac, director del Instituto Max Planck de Óptica Cuántica, ha recibido el Premio Wolf de Física 2013 junto al investigador Peter Zoller. En declaraciones a Tribuna Complutense asegura: "el premio supone una satisfacción muy grande, pues reconoce el trabajo que mis colaboradores y yo hemos realizado durante los últimos años. Es un premio con gran repercusión internacional, con lo cual potencia el campo de la información cuántica y los gases atómicos fríos". Añade que "la computación cuántica progresa de manera continua, pero todavía falta mucho tiempo para que podamos construir un ordenador cuántico lo suficientemente potente. La simulación cuántica progresa mucho más rápidamente, y ya hay experimentos que dicen que pueden hacer simulaciones que son imposibles con ordenadores clásicos". Cirac acaba de recibir además la Medalla de Honor del Instituto Niels Bohr en "reconocimiento por su contribución al desarrollo de nuevas teorías sobre el futuro de redes de información basadas en las leyes de la mecánica cuántica".

Cirac se licenció en la Facultad de Físicas de la UCM en 1988. ■

La computación en la nube se postula como herramienta contra la crisis

► POR SEGUNDO AÑO CONSECUTIVO LA REVISTA HPC IN THE CLOUD SE HA HECHO ECO DE PROYECTOS DE FIN DE CARRERA DE ESTUDIANTES DE LA FACULTAD DE INFORMÁTICA DIRIGIDOS POR EL PROFESOR JOSÉ LUIS VÁZQUEZ-POLETTI



El profesor José Luis Vázquez-Poletti con sus alumnos en la azotea de la Cadena SER

"Los alumnos son nuestro mejor activo". Es una frase habitual en boca de José Luis Vázquez-Poletti, profesor del Departamento de Arquitectura de Computadores y Automática, de la Facultad de Informática. En su caso no es sólo una frase hecha, sino que además ha conseguido, por segundo año consecutivo, que los trabajos de sus estudiantes aparezcan en la revista digital estadounidense HPC in the Cloud. A raíz de eso, incluso les llamaron de la Cadena Ser para que dieran a conocer sus trabajos.

Vázquez-Poletti informa de que "en esta ocasión, la crisis se ha convertido en el centro de gravedad de los proyectos fin de carrera. En esta ocasión son propuestas directas a la Universidad Complutense de Madrid para conseguir una provisión óptima en cuanto a costes de los recursos computacionales destinados a la docencia (CygnusCloud) y a la investigación (SmartCloud)". Los dos proyectos están codirigidos por José Antonio Martín Hernández, del mismo Departamento que Vázquez-Poletti.

Adrian Fernandez, Samuel Guayervas y Luis Barrios son los responsables de CygnusCloud. El objetivo de este proyecto es que se pueda acceder a todo tipo de aplicaciones desde cualquier ordenador de la UCM, utilizándolas desde la nube. Es decir, sin necesidad de que la Complutense haga una importante y cara inversión en todo tipo de programas.

SMARTCLOUD Y PYMES

César Cayo, Javier Bachrachas y Ailyn Baltá están detrás de SmartCloud. En este caso se busca rentabilizar todos los ordenadores que existen en el campus cuando no están utilizándose por motivos como el horario y cuyo uso podría beneficiar a los investigadores complutenses.

Vázquez-Poletti dirige además un proyecto de Guillermo Marco, estudiante del Magíster de Bioinformática y Biología Computacional. Este proyecto bautizado Horadrim propone una solución a otro sector que también sufre la crisis: el de las PYMES del sector tecnológico más puntero. ■

34





12.3.3. ARTÍCULO EN LA FACULTAD DE INFORMÁTICA, UCM

Alumnos de la Facultad de Informática son entrevistados en la Cadena SER

El martes 15 de Enero, la Cadena SER entrevistó en su programación matinal a los alumnos de la Facultad Luis Barrios y Javier Bachrachas, junto a su profesor José Luis Vázquez-Poletti, para conocer de primera mano los proyectos fin de carrera que en sus respectivos grupos se están realizando durante el presente curso académico.



El profesor José Luis Vázquez-Poletti rodeado de sus alumnos en la azotea de la Cadena SER.

Estos dos proyectos, codirigidos por el también profesor de la Facultad José Antonio Martín H., tienen dos aspectos en común. El primero es la tecnología empleada: la Computación en la Nube (Cloud Computing), ya que gracias a su gran accesibilidad y proceso rápido de aprendizaje, los alumnos tienen acceso a todo un mundo de posibilidades que les permite desarrollar todo tipo de proyectos interesantes en un tiempo muy razonable.



Durante la entrevista.

La segunda es la voluntad por parte de los dos grupos de alumnos de querer aportar su granito de arena a la Universidad Complutense de Madrid en la situación de crisis actual. En particular, los dos proyectos son propuestas directas para conseguir una provisión óptima en cuanto a costes de los recursos computacionales destinados a la Docencia y a la Investigación.

	<p>OptimCloud Adrián Fernández, Samuel Guayherbas y Luis Barrios</p> <p>Este proyecto propone la provisión puestas de laboratorio virtuales que puedan ser accedidas desde cualquier PC del campus de la UCM. De esta manera, se reducirían los efectos de los recursos en equipo informático al uso, ya que los estudiantes podrían usar ordenadores más baratos, reduciendo el consumo de electricidad.</p> <p>Por otro lado, se optimizaría el uso de las salas de informática ya especializadas repartidas por todo el campus, además de reduciendo los costes totales ya que OptimCloud está desarrollado enteramente con Software Libre.</p> <p>Por si fuera poco, estos alumnos han llevado su proyecto al Concurso Universitario de Software Libre (http://www.concursosoftwarelibre.org/2012/).</p>
	<p>SmartCloud Óscar Gajo, Javier Bachrachas y Adijet Sahli</p> <p>El objetivo de este proyecto es exprimir al máximo el valor del parque computacional de la UCM que, por nosotros como el horario, está infrautilizado y cuyo uso podría beneficiar enormemente a los investigadores computacionales.</p> <p>Como ordenadores infrautilizados se tiene que proporcionar necesariamente a laboratorios o salas de informática. También aquellos empleados por el cuerpo administrativo pueden ser utilizados en esta solución. Por ello, el software llamado las máquinas donadas de forma temporal se utilizan para no interferir en su correcta funcionamiento.</p> <p>Uno de los servicios ofrecidos por SmartCloud es un sistema distribuido tipo MediaB basado en IPython, de gran aceptación en la comunidad científica.</p>

Descripción de los proyectos fin de carrera.

El audio de la entrevista puede ser accedido en la siguiente dirección:
http://www.cadenaser.com/cultura/audios/noticias-madrid-charlamos-universidad/cscrcsppor/20130115cscrcsrucl_8/Aes/





Plataforma de Computación Científica sobre Infraestructura Cloud Privada Oportunista

Javier Bachrachas Peterburg, Ailyn Baltá Camejo, César Cayo Ventura, José Luis Vázquez-Poletti y José Antonio Martín H.

Departamento de Arquitectura de Computadores y Automática, Facultad de Informática, Universidad Complutense de Madrid, 28040, Madrid (España)

Abstract—SmartCloud es una solución completa (IaaS + SaaS) siguiendo el paradigma de cloud computing. A través de una arquitectura de capas escalable se implementa una nube privada que gestiona los elementos de una infraestructura ya existente añadiendo a los mismos la funcionalidad de nodo de procesamiento en los tiempos de inactividad sin afectar su utilización habitual. Sobre esta plataforma se ha implementado una solución de cálculo científico open source (iPython) que utiliza los recursos antes mencionados como soporte hardware para procesamiento paralelo y funcionalidades de clúster. Como resultado de esta solución se puede brindar a los usuarios un nuevo servicio sin inversión en hardware ni licencias con el valor añadido de la reutilización de infraestructuras ya existentes.

Palabras clave—Cloud Computing, iPython, reutilización hardware, optimización infraestructuras, educación.

I. INTRODUCCIÓN

LA situación económica actual ha hecho que muchos grupos de trabajo en informática enfoquen sus esfuerzos en encontrar nuevas maneras de optimizar recursos o ampliar los servicios brindados a los usuarios sin que esto implique un aumento en los costes[8].

El primer enfoque que surge ante esta necesidad es migrar los sistemas hacia el software de código abierto, algo que es tendencia actual en organizaciones tanto privadas como estatales[4]. Un ejemplo de esto es la decisión del gobierno de Extremadura de migrar sus sistemas a código abierto y Linux, con un ahorro estimado de 30 millones de euros anuales[3].

En términos de la capa de aplicación, en el mundo de la investigación y la enseñanza también existen alternativas de código abierto a herramientas bien establecidas en el mercado como Matlab o Mathematica, las que al ser un software propietario tienen coste de licencia y su desarrollo está cerrado a una compañía, impidiendo las mejoras de la comunidad. En este proyecto hemos realizado una implantación de iPython[1] como alternativa a las aplicaciones antes mencionadas, permitiendo lograr una funcionalidad similar en términos de potencia de cálculo, pero adaptada a las necesidades y paradigma de este proyecto.

Un paso más en el análisis del problema, nos lleva a plantear la pregunta de cómo optimizar el soporte hardware de las aplicaciones. Nuestra aproximación a este problema es la reutilización de infraestructura ya existente para aprovechar su capacidad de procesamiento en los tiempos de inactividad[6].

Tomando como ejemplo la situación de la Facultad de Informática de la Universidad Complutense de Madrid, existen grupos de investigación que necesitan realizar procesamiento

de datos y como la Facultad no puede brindarle servicios de procesamiento (o bien por qué no los tiene, o por restricciones presupuestarias), deben recurrir a proveedores externos como por ejemplo Amazon¹.

Nuestra solución a este problema es utilizar los ordenadores de los laboratorios de la Facultad cuando no están siendo utilizados, y sobre cada uno de ellos iniciar una máquina virtual con un nodo de procesamiento, lo que tiene un impacto nulo en la utilización rutinaria de los recursos y nos permite generar un clúster dinámico de procesamiento[9].

Hay que recordar los problemas que conlleva utilizar proveedores externos para el procesamiento de datos en términos de latencia y de protección de datos[21], por lo que hemos creado la solución “SmartCloud” como una propuesta para optimizar recursos y lograr brindar un mejor servicio a usuarios tal y como se puede apreciar en la Figura 1.

La solución se trata de una plataforma de Cloud Computing de nube privada² pues se lleva a cabo con los propios recursos existentes (máquinas, redes, almacenamiento, instalaciones) y su acceso está restringido a una red cerrada. Incluye un componente de Infrastructure as a Service (IaaS) y un componente de Software as a Service (SaaS) unidas por una capa de integración.

Se trata de una solución escalable y flexible, dado que al contar con un diseño arquitectónico por capas, se puede reutilizar o modificar la plataforma para implementar otras

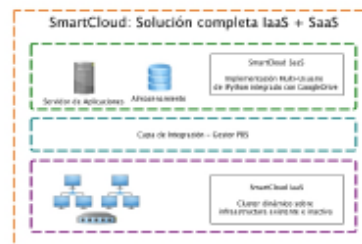


Figura 1: Diagrama de Solución SmartCloud integrando IaaS + SaaS

¹<http://aws.amazon.com/es/ec2/>

²<http://www.postgradoeinvestigacion.uadec.mx/CienciaCierta/CC29/11.html>





soluciones software sobre la misma.

El servicio brindado al usuario final se trata de una implementación de iPython multiusuario, que permite que cualquier usuario del sistema acceda a través de un navegador de internet a nuestra aplicación, permitiéndole realizar simulaciones, gráficos, cálculos, procesamiento de código Python y otras funcionalidades con la ventaja del procesamiento distribuido. El procesamiento no se realiza en el ordenador del cliente sino en nuestro servidor centralizado y en el clúster dinámico antes mencionado.

Nuestra plataforma permite adicionalmente la sincronización de archivos de usuario con Google Drive, permitiendo el trabajo colaborativo.

En esta contribución primero se estudiará el estado del arte para evaluar las alternativas existentes. Luego, se explicará la arquitectura de la solución haciendo la distinción entre diseño de alto y bajo nivel. Posteriormente se detallará el modelo de ahorro de costes e impacto económico de la solución, lo que ayuda a entender la motivación del proyecto. Para terminar se explicarán las conclusiones del trabajo y las posibles ampliaciones que puede tener.

II. ESTADO DEL ARTE

Con el objetivo de implementar una solución siguiendo el paradigma de Cloud-Computing, hemos basado nuestra aproximación en aplicaciones ya existentes que cuentan con una buena aceptación en la comunidad de usuarios y han demostrado buenos resultados. Se han valorado los casos particulares de Boinc³ y Wolfram — Alpha⁴.

La Infraestructura Abierta de Berkeley para la Computación en Red (BOINC) es una infraestructura para la computación distribuida, diseñada para ser una plataforma libre. Está dirigido a proyectos que requieren una gran capacidad de cálculo a través de la utilización de ordenadores personales distribuidos alrededor del mundo, realizando los cálculos en los ciclos de CPU o GPU que no son utilizados. BOINC consiste en un servidor y un cliente que se comunican para distribuir y procesar el trabajo. Cada ordenador es asignado a un único proyecto⁵. Wolfram—Alpha tiene la apariencia de motor de búsqueda pero no busca respuestas a las preguntas en un conglomerado de páginas web o documentos, sino que realiza cálculos en su servidor y envía la respuesta para que sea visualizada por el usuario a través de una interfaz web⁶.

Se concluyó que no existía una solución completa que proporcionara los servicios que se querían ofrecer pero si existían soluciones que los cubrían parcialmente, por lo que se realizó un análisis de las herramientas existentes como componentes para implementar una solución completa.

Como parte del diseño arquitectónico de la aplicación surge la necesidad de utilizar: un hipervisor, un gestor de colas PBS para la capa de integración y una herramienta de cálculo potente con interfaz web.

La versión analizada de VirtualBox es de código abierto y sujeto a la licencia GPL⁶. Además, es funcionalmente idéntico

en todas las plataformas de acogida y utiliza el mismo archivo y formato de imagen. Esto permite ejecutar la misma imagen de máquina virtual en instancias de VirtualBox que se ejecutan sobre diferentes sistemas operativos por lo que es fácil exportar las imágenes entre entornos, ofreciendo mayor portabilidad. VirtualBox está en constante desarrollo y actualización por lo que nuevas funcionalidades son implementadas regularmente. Además, cuenta con una API que nos permite integrarlo fácilmente a nuestra solución⁶.

VMware también cuenta con una versión gratis con funcionalidades similares a VirtualBox, pero al realizar una comparativa de las tecnologías se descarta dado que la filosofía de la solución es utilizar software de código abierto⁷.

El gestor PBS Torque (Tera-scale Open-source Resource and Queue manager) es un administrador de recursos que nos provee de mecanismos para controlar trabajos sobre distintos nodos de un clúster. Permite gestionar el estado de los nodos y su utilización. Basado en OpenPBS, es el sistema de colas batch de código abierto líder en el mercado. Soporta el procesamiento en paralelo de múltiples batches y colas de trabajo interactivas⁸.

Otro de los objetivos de la solución es ofrecer una interfaz web. El estudio de herramientas disponibles se enfocó en encontrar un entorno interactivo que se pudiera integrar a la infraestructura existente y que contara con soporte multiusuario, pero que a la vez fuera una poderosa herramienta de cálculo. Considerando las restricciones mencionadas se evaluaron las alternativas de Sage y iPython.

Sage fue diseñado desde un principio como solución escalable a grandes servidores multiusuario centralizados (ejemplo sagenh.org con cerca de 76000 cuentas), pero con la desventaja de que el mismo binario que se ejecuta en grandes servidores es utilizado para ejecuciones que no requieren tantos recursos, lo que se refleja en un bajo rendimiento en ciertos entornos. Sage tiene licencia GPL⁹. IPython es multiplataforma, software libre, tiene una enorme comunidad detrás, un desarrollo constante y bien organizado, permite paralelismo y es extremadamente potente. Nació de las manos de investigadores y en un ámbito científico, por lo que su comunidad es mayormente académica al igual que nuestros usuarios. IPython trabaja con un sistema de archivos standard accesible trivialmente desde el sistema operativo, facilitando la realización de operaciones rutinarias en el análisis de datos y simulación numérica como acceder a scripts locales o ver archivos arbitrarios. IPython incluye una serie de opciones para dar formato a los documentos: capacidades multi-hoja, preámbulo de estilo LaTeX, metadatos por celda, navegación outline-level, y apoyo validación / reproducibilidad entre otras. IPython tiene licencia BSD. Además, permite añadir nuevas extensiones como Matlab, SQL y Physicsy¹⁰.

³<http://boinc.berkeley.edu/>

⁴<http://www.wolframalpha.com/>

⁵<http://es.wikipedia.org>

⁶<http://www.virtualbox.org/>

⁷<http://www.vmware.com/es/>

⁸<http://www.adaptivecomputing.com/products/open-source/torque/>

⁹<http://www.sagemath.org/>

¹⁰<http://github.com/ipython/ipython/wiki/Extensions-Index#matlab>





III. ARQUITECTURA DE LA SOLUCIÓN

A. Diseño de alto nivel

Después de un análisis en profundidad llegó a la conclusión de implementar una arquitectura por capas por su beneficio modular.

- Capa Inferior: formada por los nodos de procesamiento conectados a una red para poder ser accesibles y gestionables por el servidor.
- Capa Intermedia de integración: comunica la capa superior con la inferior. Gestiona la distribución de los trabajos generados desde la capa superior.
- Capa Superior: contiene el front-end para la interacción con el usuario. En esta capa se encuentra la aplicación de cálculo que se despliega. Cuando la herramienta necesite realizar funciones sobre el cluster realizará llamadas a capas inferiores, lo que se refleja en un aumento en la carga del procesamiento del cluster.

Dada a la diversidad de alternativas para implementar estas capas, con el previo análisis de cada una, se llegó a las conclusiones de diseño:

- En la capa inferior para definir los nodos de procesamiento se decidió realizarlo mediante virtualización para homogeneizar las máquinas a usar. Como ya se ha comentado anteriormente, en cada nodo se instala una máquina virtual y sobre ésta una misma imagen. Se decidió utilizar Oracle VM VirtualBox, un hipervisor de tipo 2, dada su facilidad de instalación y configuración y sus garantías de acceso al hardware físico a través del sistema operativo host. Para el sistema operativo a virtualizar se decidió Ubuntu (Linux) por ser software libre. Ver figura 2.
- Para la capa intermedia, se decidió utilizar un gestor de trabajos PBS Torque que nos permite una gestión de procesamiento transparente. Además es el sistema de colas batch de código abierto líder en el mercado.
- Para la capa superior, se eligió iPython gracias a su entorno completo para la computación interactiva y exploratoria. Además, presenta una arquitectura robusta construida alrededor del concepto de intérprete de Python pero a la vez sencilla y eficaz[2]. También se tuvo en cuenta el interés creciente en esta herramienta (Figura 3).



Figura 2: Arquitectura de capas - Capa inferior

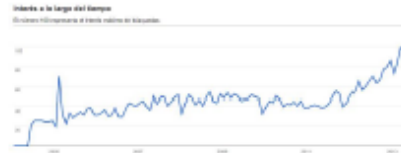


Figura 3: Evolución del interés en iPython

B. Diseño de bajo nivel

Para la capa inferior fue necesario añadir tareas al SO que inicien la máquina virtual cuando detecte que no se estaba utilizando, la apaguen cuando reconociera un inicio de sesión e informen de su estado al cluster. En las máquinas virtuales se instala una imagen de Ubuntu 10.04 debido a su facilidad de administración.

Como capa intermedia se utiliza el gestor de trabajos Torque. Se ha definido una única cola en esta herramienta en la que convergen todas las peticiones de las instancias de iPython. Al utilizar un gestor de trabajos, se simplifica la integración de ambas capas dado que es una interfaz transparente para la capa superior. En otras palabras, iPython solamente ve un único punto donde realizar peticiones/envíos de trabajos, aunque éste se encarga de distribuir los trabajos entre los recursos disponibles según políticas de optimización.

En la capa superior se define un frontend con gestión de usuarios. Para facilitar esta gestión y no guardar información de la cuenta de los usuarios, se utiliza la openID de Google y se evalúa si el usuario autenticado tiene permisos para ejecutar el notebook de iPython. Inicialmente iPython no estaba desarrollado para poder ser utilizado por varios usuarios de manera simultánea bajo un mismo servidor, razón por la cual tuvimos que modificarlo para añadir esta funcionalidad. Dado que un usuario utiliza la cuenta de Google, tiene la posibilidad de sincronizar sus hojas con Google Drive. La figura 4 ilustra la arquitectura de la aplicación

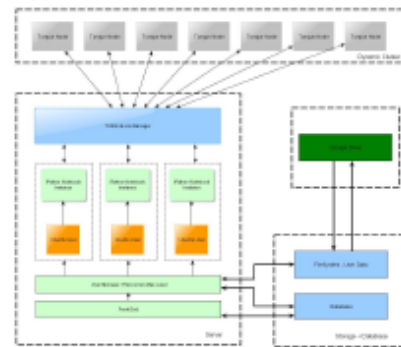


Figura 4: Arquitectura de la solución





IV. ESTUDIO DE COSTES E IMPACTO ECONÓMICO

Uno de los objetivos de SmartCloud es lograr optimizar la utilización de la infraestructura existente, lo que se refleja en un ahorro en costes. Actualmente si la universidad necesita realizar cálculos que exceden sus capacidades de procesamiento debe recurrir a un proveedor externo[5]. Al utilizar el cluster dinámico de SmartCloud, se pueden ofrecer los mismos servicios de procesamiento sin tener que contratar a un proveedor externo y sin la latencia que implica el procesamiento remoto. El único sobrecoste significativo que implica esta plataforma, es el coste energético[10]. A continuación se detalla el modelo económico desarrollado para realizar estos cálculos[11].

$$C = W_l * T * F_c * F_u * C_e$$

$$O = \frac{W_l * (T_b + F_c * F_u * T_i) + W_i * (T_i * (1 - F_c * F_u))}{W_l * T_b + W_i * T_i}$$

En donde términos de costes C es el coste energético final y O el sobrecoste energético derivado de utilizar SmartCloud. W_l es el consumo energético de cada nodo con una carga standard en Kw/h y W_i el consumo energético de cada nodo estado inactivo en Kw/h. T se refiere al tiempo disponible para el procesamiento, T_b al Tiempo de utilización del clúster con carga sin utilizar SmartCloud, y T_i al tiempo de utilización del clúster en estado inactivo sin utilizar SmartCloud. Para estimar la carga se usan las variables F_c como factor de carga del clúster en el tiempo disponible, y F_u como factor de utilización de las horas disponibles. El coste de la energía en Kw/h se expresa con C_e .

Se plantea un escenario de ejemplo para estudiar el problema. Se dispone de 20 ordenadores i7-3770K con 4GB de RAM y 4 cores con hyper-threading, en un laboratorio con la distribución de uso indicada en la figura 5.

En una semana hay 118 horas inactivas que podrían ser utilizadas por SmartCloud, formando un cluster de 160 cores con 80 GB de RAM. Supongamos que se necesita utilizar todo el cluster al 100 %

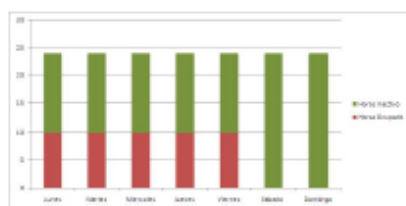


Figura 5: Distribución de uso de ordenadores en el caso de estudio.

Cuadro I: Comparación de proveedores Cloud

Server Type	CPU	RAM	I/O	Base Hourly Cost (USD)	Hourly Cost (USD)	Coste Mensual
HP	16xCPU	8TB	120,000	0.14	0.38 €	1,021,987 €
Backspace	16xCPU	8TB	100,000	0.22	0.60 €	1,580,074 €
SoftLayer	16x1.0GHz	7.5GB	0.000	0.23	0.70 €	1,870,278 €
AWS	16x1.0GHz	7.5GB	800,000	0.24	0.76 €	1,731,209 €

Cuadro II: Latencias de Amazon desde España.

Service	TimeLat	# of Samples	Min ms	Max ms	Std Dev	Median	Avg ms
Amazon EC2	1.13	3	231	208	13.2 %	208	272.33
Amazon CloudFront	0.07	6	62	96	15.0 %	78	75.5

de la carga, en el total de las horas disponibles. El consumo energético sería de 196,6€ lo que presenta un sobrecoste de 50 % respecto al consumo de energía actual. Ahora bien, si se recurriera a proveedores externos para realizar este procesamiento, esto tendría un coste de media de 1508,68€. El cuadro I muestra una comparación de proveedores de servicios cloud que proporcionan un servicio equivalente ¹¹.

Para calcular el ahorro final simplemente basta con restar al coste externo el coste energético, lo que se refleja en 1312€ de ahorro¹¹. En términos de latencias, el trabajo dentro de una red local como lo que propone SmartCloud implica latencias sensiblemente menores frente a las que se experimentan trabajando con proveedores cloud. El cuadro II muestra las latencias de Amazon desde España¹².

Claramente son mayores a las latencias dentro de una red local. A su vez, iPython es código abierto, lo que comparado con herramientas como Matlab o Mathematica que tienen costes de licencias representa un ahorro significativo.

V. CONCLUSIONES Y TRABAJO FUTURO

La importancia que tiene la utilización de plataformas cloud se refleja en el ahorro de costes tanto energéticos como en infraestructura tal como se refleja en el trabajo actual.

Uno de los principales aportes de este trabajo es la posibilidad de utilizar iPython en modo multiusuario, ya que actualmente la herramienta no implementa esta funcionalidad.

Al utilizar esta solución, se ofrece a los investigadores no sólo una poderosa herramienta de cálculo sino también una manera de aprovechar los recursos que existen sin costes significativos añadidos. Además, la integración de la herramienta con Google Drive facilita mucho más su usabilidad.

La plataforma resultante presenta una arquitectura de cluster. Una vez implementada, permite el despliegue de otras herramientas y recursos como apoyo a otros proyectos, nuevas aplicaciones, procesamiento batch, entre otros.

Este proyecto puede continuar para seguir explotando la escalabilidad e integración con otros recursos. Se puede dar la posibilidad de integrarlo con otros servicios cloud por ejemplo los de Amazon ¹³. El diseño de una nueva solución

¹¹ <http://planforcloud.rightscale.com>

¹² <http://www.cloudharmony.com/>

¹³ <http://gd.msu.edu/angus/beacon-2012/week1.html>





se puede realizar desde distintos enfoques dado que se adapta a las tendencias actuales de cloud, permitiendo una mayor versatilidad¹⁴.

Para este proyecto se ha desplegado en el cluster una herramienta en concreto, Ipython, pero cualquier aplicación distribuida podría ser desplegada, por ejemplo específicas de un proyecto o de alguna empresa que en determinado momento tenga un pico alto de consumo de recursos.

Una vez logrado el objetivo principal y consolidada la infraestructura, se podría ofrecer estos servicios de cálculo a otras facultades o entidades externas[7].

Como posible ampliación, se puede trabajar en un modelo de cluster híbrido con máquinas virtuales, servidores físicos y balanceadores de carga como alternativa a la actual implementación. También se pueden iniciar otras imágenes sobre las máquinas virtuales para implementar nuevas funcionalidades.

Un estudio en profundidad en lo referente a seguridad permitiría resolver los posibles fallos o vulnerabilidades del sistema, en el actual desarrollo solo se cubren los estándares mínimos necesarios.

REFERENCIAS

- [1] E. Pérez. "The Ipython notebook: a historical retrospective", 2012
- [2] E. Pérez, B. Granger, M. Ragan-Kelley. "Ipython: a very quick overview", http://perez.org/talks/1010_ipython_overview.pdf
- [3] CIO. "Extremadura prevé ahorrar 30 millones de euros al año con diez proyectos de software libre", <http://www.ciopain.es/aapp/extremadura-preve-ahorrar-30-millones-de-euros-al-ano-con-diez-proyectos-de-software-libre>
- [4] A. Bernardo. "Cuáles son las universidades españolas que apoyan el software libre en 2013?", <http://alt1040.com/2013/05/universidades-espanolas-que-apoyan-el-software-libre-2>
- [5] D. Kondo, B. Javadi, P. Malecot, F. Cappello, D.P. Anderson. "Cost-benefit analysis of Cloud Computing versus desktop grids", *Parallel & Distributed Processing*, Addison-Wesley, 1989.
- [6] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, A. Ghalsasi. "Cloud computing - The business perspective", *Decision Support Systems*, Volume 51, Issue 1, April 2011, Pages 176-189
- [7] R. Buyya, C. Shin Yeo, S. Venugopal, J. Broberg, I. Brandic. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", *Future Generation Computer Systems*, Volume 25, Issue 6 June 2009, Pages 599-616
- [8] F. Etro. "The Economic Impact of Cloud Computing on Business Creation, Employment and Output in Europe: An application of the Endogenous Market Structures Approach to a GPT innovation", *Review of Business and Economics, Katholieke Universiteit Leuven, Faculteit Economie en Bedrijfswetenschappen* vol. 0(2), pages 179-208.
- [9] N. Sultan. "Cloud computing for education: A new dawn?", *International Journal of Information Management*, Volume 30, Issue 2, April 2010, Pages 109-116
- [10] A. Bert, E. Gelenbe, M. di Girolamo, G. Giuliani, H. de Meer, M. Quan Dang, K. Pentikousis. "Energy-Efficient Cloud Computing", *The Computer Journal*, Number 7, 2010, Volume 53
- [11] G. D. Guerrero, R. M. Wallace, J. L. Vázquez-Poletti, J. M. Cecilia, J. M. García, D. Mazos and H. Pérez-Sánchez. "Concurrency and Computation: Practice and Experience, Special Issue on Distributed, Parallel and GPU-accelerated Approaches to Computational Biology" 2013, In Press

¹⁴<http://aws.amazon.com/elastic/>





13. GLOSARIO

Sistema operativo del host (host OS): Este es el sistema operativo del equipo físico en el que se ha instalado VirtualBox.

Sistema operativo invitado (guest OS): Este es el sistema operativo que se ejecuta en la máquina virtual.

Virtualización: Es la creación -a través de software- de una versión virtual de algún recurso tecnológico, como puede ser una plataforma de hardware, un sistema operativo, un dispositivo de almacenamiento u otros recursos de red.

Máquina virtual: Es un software que simula a una computadora y puede ejecutar programas como si fuese una computadora real.

Cloud computing: es un nuevo modelo de prestación de servicios de negocio y tecnología, que permite incluso al usuario acceder a un catálogo de servicios estandarizados y responder con ellos a las necesidades de negocio

SaaS: Es un modelo de distribución de software donde el soporte lógico y los datos que maneja se alojan en servidores de una compañía de tecnologías de información y comunicación (TIC), a los que se accede con un navegador web desde un cliente, a través de Internet.

IaaS: Esta en la capa inferior y es un medio de entregar almacenamiento básico y capacidades de cómputo como servicios estandarizados en la red. Servidores, sistemas de almacenamiento, conexiones, enrutadores, y otros sistemas se concentran.

Nube privada: Se define como la instalación de una red informática propietaria o un centro de procesamiento de datos que abastece de servicios alojados por un número limitado de usuarios.

Nube pública: es un término que hace referencia a una "infraestructura que está disponible para el público en general o para un gran sector de la industria, y es provista por una organización que comercializa servicios a demanda

Nube híbrida: Se define como combinación de los modelos de nubes públicas y privadas. Las nubes híbridas ofrecen la promesa del escalado aprovisionada





externamente, en-demanda, pero añaden la complejidad de determinar cómo distribuir las aplicaciones a través de estos ambientes diferentes.

Network file system (nfs): es un protocolo de nivel de aplicación, según el Modelo OSI. Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local.

Portable Batch System: es un sistema flexible de balanceo de carga y planificación de tareas, inicialmente fue desarrollado para administrar recursos computacionales de la NASA.

Secure Shell (SSH): Es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos.

RSA (Rivest, Shamir y Adleman) : Es un sistema criptográfico de clave pública desarrollado en 1977. Es el primer y más utilizado algoritmo de este tipo y es válido tanto para cifrar como para firmar digitalmente.

Web Server: Servidor de aplicaciones web

Local Área Network: Es la interconexión de una o varias computadoras y periféricos. Antiguamente su extensión estaba limitada físicamente a un edificio o a un entorno de 200 metros.

Grid: Es una tecnología innovadora que permite utilizar de forma coordinada todo tipo de recursos (entre ellos cómputo, almacenamiento y aplicaciones específicas) que no están sujetos a un control centralizado.

iPython: Es un shell interactivo que añade funcionalidades extra al modo interactivo incluido con Python, como resaltado de líneas y errores mediante colores, una sintaxis adicional para el shell, autocompletado mediante tabulador de variables, módulos y atributos; entre otras funcionalidades.

IPython Notebook: Es un entorno computacional interactivo basado en la web donde se puede combinar la ejecución de código, texto, matemáticas en un solo documento.

Python: Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible.

Frontend: Es la parte de un sistema de software que interactúa directamente con el usuario.

Hipervisor: Es una plataforma que permite aplicar diversas técnicas de control de virtualización para utilizar, al mismo tiempo, diferentes sistemas operativos.





SmartCloud: Gestión Eficiente de Recursos Computacionales de la Administración Pública

Linux: Sistema operativo de código abierto.

Google Drive: Es un servicio de almacenamiento de archivos en línea. Google Drive es un reemplazo de Google Docs que ha cambiado su dirección de enlace de docs.google.com por drive.google.com entre otras cualidades.





14. REFERENCIAS

Bibliografía:

- [1] F. Pérez ' "The IPython notebook: a historical retrospective", 2012
- [2] F. Pérez, ' B. Granger, M. Ragan-Kelley, "IPython: a very quickoverview", http://fperez.org/talks/1010_iPython_overview.pdf
- [3] CIO, "Extremadura prevee' ahorrar 30 millones de euros al año~ con diez proyectos de software libre", <http://www.ciospain.es/aapp/extremadura-preve-ahorrar-30-millones-de-euros-al-ano-con-diez-proyectos-de-software-libre>
- [4] A. Bernardo, "Cuales' son las universidades españolas~ que apoyan el software libre en 2013?", <http://alt1040.com/2013/05/universidades-espanolas-que-apoyan-el-software-libre-2>
- [5] D. Kondo, B. Javadi, P. Malecot, F. Cappello, D.P. Anderson, "Cost-benefit analysis of Cloud Computing versus desktop grids", Parallel & Distributed Processing, Addison-Wesley, 1989.
- [6] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, A. Ghalsasi, "Cloudcomputing - The business perspective", Decision Support Systems, Volume 51, Issue 1, April 2011, Pages 176-189
- [7] R. Buyya, C. Shin Yeo, S. Venugopal, J. Broberg, I. Brandic "Cloudcomputing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, Volume 25, Issue 6 June 2009, Pages 599-616
- [8] F. Etro "The Economic Impact of Cloud Computing on BusinessCreation, Employment and Output in Europe. An application of the Endogenous Market Structures Approach to a GPT innovation", Review of Business and Economics, Katholieke Universiteit Leuven, Faculteit Economie en Bedrijfswetenschappen vol. 0(2), pages 179-208.
- [9] N. Sultan , "Cloud computing for education: A new dawn?", International Journal of Information Management, , Volume 30, Issue2, April 2010, Pages 109-116
- [10] A. Berl, E. Gelenbe, M. di Girolamo, G. Giuliani, H. de Meer, M. Quan Dang, K. Pentikousis "Energy-Efficient Cloud Computing", TheComputer Journal , Number 7, 2010, Volume 53
- [11] G. D. Guerrero , R. M. Wallace, J. L. Vazquez' -Poletti, J. M. Cecilia, J. M. García, D. Mozos and H. Perez-Sanchez' , "Concurrency andComputation: Practice and Experience, Special Issue on Distributed, Parallel, and GPU-accelerated Approaches to Computational Biology" 2013, In Press
- [12] <http://www.adaptivecomputing.com/products/open-source/torque/>





- [13] <http://planforcloud.rightscale.com/>
- [14] <http://cloudharmony.com>
- [15] <http://es.wikipedia.org>
- [16] <http://guerratorpia.com/instalar-openssh-server-en-ubuntu>
- [17] <http://www.ehu.es/ehusfera/ghym/2010/10/15/identificacion-automatica-en-ssh-usando-claves-rsa/>
- [18] <http://www.clusterresources.com/torquedocs21/>
- [19] <http://www.google.es/trends/?hl=es>
- [20] <http://www.statista.com/statistics/13020/electricity-prices-in-selected-countries/>
- [21] <http://eloy-mp.com/wordpress262/?p=1851>



