1366: Gestión sostenible de clústers de recursos virtuales

Memoria del proyecto

de Ingeniería en Informática

realizado por

Enric Pagès Montanera

y dirigido por

Vicente José Ivars Camañez,

Josep Martrat Sotil y

Ana Juan Ferrer

Bellaterra, 18 de septiembre de 2009

El subasignado, *Vicente José Ivars Camañez*

Profesor de la Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el trabajo al que corresponde esta memoria ha sido realizado bajo su dirección por

Enric Pagès Montanera

Y para que conste la firma presente.

Vicente José Ivars Camañez
Bellaterra, 18 de septiembre de 2009

El subasignado, *Josep Martrat Sotil*

Head of Unit de Service-Oriented Middleware and Infrastructure en Atos Origin,

CERTIFICA:

Que el trabajo al que corresponde esta memoria ha sido realizado bajo su dirección por

Enric Pagès Montanera

Y para que conste la firma presente.

Josep Martrat Sotil
Barcelona, 18 de septiembre de 2009

A mis padres, hermana y pareja por el apoyo incondicional desde el inicio.

Agradecimientos

Me gustaría agradecer el esfuerzo, inspiración y continuo *feedback* de mi *nephele* y musa de las nubes, Ana Juan Ferrer que me ha empujado hacia adelante para concluir esta ardua tarea.

Un recordatorio especial para mi *Lord of the systems* particular, Javier. Gracias por arrimar el hombro en los tediosos momentos de depuración de la solución, enmarañados entre *logs* interminables.

Agradecer también, tanto a Vicente Ivars como a Josep Martrat por el saber hacer con el que han dirigido el proyecto, facilitando enormemente las tareas del día a día.

Destacar al departamento de Atos Research & Innovatión por proporcionar los recursos necesarios y a todos mis compañeros de unidad por el apoyo continuado durante estos meses. También a los compañeros de universidad que me han acompañado durante el largo viaje que supone una ingeniería superior.

Este proyecto tiene como finalidad ofrecer un servicio de computación en forma de máquina virtual, utilizando los recursos internos de Atos Research & Innovation. Además, se pretende implementar este servicio sobre los excendentes de máquinas del propio departamento. La prestación de este servicio se realiza mediante un gestor de la infraestructura de forma centralizada. Para la implantación de este entorno se ha definido la adopción en fases y profundizado en dos de los gestores más activos en la investigación del modelo Cloud Computing (Open Nebula, Eucalyptus).

This project aims to offer a computational service as a virtual machine, using the Atos Research and Innovation's internal resources. Furthermore, it aims to implement this service upon the surplus machines of this department. The delivery of this service will be carried out using a centralised management infrastructure. To achieve this end, adoption according to phases has been defined and in depth investigation of the two management tools most used in cloud computing reserch has been carried out (Open Nebula, Eucalyptus).

Aquest projecte te com a finalitat oferir un servei computacional en forma de máquina vitual, utilitzant els recursos interns de Atos Reseach & Innovation. A més, es preten implementar aquest servei sobre les màquines excedents d'aquest departament. La prestació del servei es realitza mitjantçant un gestor per a la infraestructura de forma centralitzada. Per a la implantació d'aquest entorn s'ha definit l'adopció en fases y profunditzat en dos dels gestors més actius en l'investigació del módel Cloud Computing (Open Nebula, Eucalyptus).

Índice

Índice de figuras

Índice de cuadros

1. Introducción

Actualmente España, Europa y el mundo en general están sumergidos en una crisis financiera que afecta a todos los sectores. No obstante, existen reconocidas teorías económicas como *The Long Waves in Economic Live* [1], del profesor Nickolai Kondratieff , que describen estos fenómenos de depresión (Winter) como períodos de limpieza que permiten reajustar la economía de los últimos excesos y comenzar a planificar las bases de recuperación y crecimiento futuros. La innovación permite redefinir las tecnologías del pasado período de crecimiento, con el objetivo de abaratar costes y conseguir implantar en el mercado estas innovaciones.

El modelo Cloud Computing y la virtualización de servidores son probablemente dos de las tecnologías emergentes que transformaran la visión de las infraestructuras en las organizciones actuales. El estudio realizado por Gartner en 2008 [figura 1] sitúa al Cloud Computing en pleno crecimiento hacia el límete de sus expectativas. Este período viene acompañado de ruido alrededor de la tecnología, que distorsiona sus capacidades reales y de los esfuerzos de la comunidad científica y empresarial por delimitar los escenarios de aplicación y clarificar su desarrollo. Inicialmente surge como modelo de comportamiento de la infraestructura como un servicio (IaaS) para evolucionar y extenderse a ofrecer todo como un servicio (XaaS).

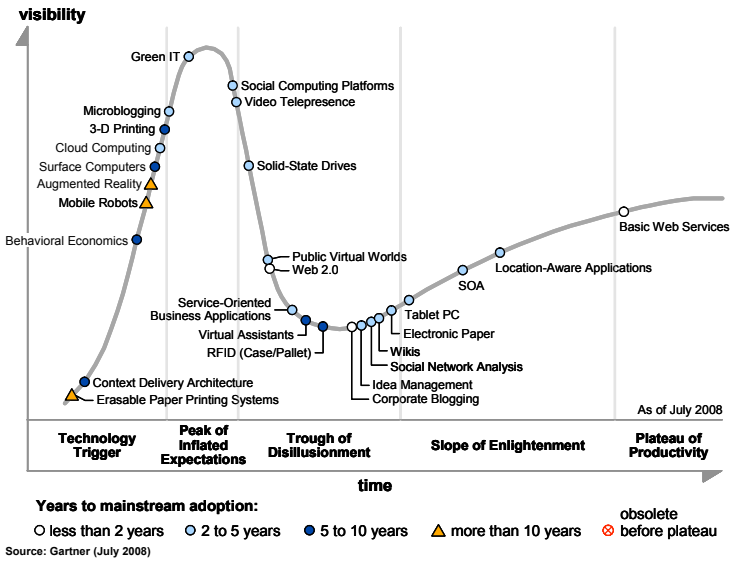


Figura 1: Hyper-ciclo de Gartner para las tecnologías emergentes 2008

La virtualización, ejerce como tecnología clave para la prestación de estos servicios, permitiendo desacoplar los recursos físicos de una infraestructura y tratarlos como software a través de la red. La virtualización de servidores permite ejecutar múltiples sistemas virtuales en un único sistema físico. Me-

diante el reaprovechamiento de los recursos se consigue reducir el coste en hardware de la infraestructura, así como el consumo eléctrico y de refrigeración. Aunque no se trata de una tecnología reciente, el estado de madurez en que se encuentra gracias al desarrollo de hipervisores eficientes, hace a la comunidad científica y empresarial estar de acuerdo en la necesidad de trabajar por una adopción viable de este modelo.

El objetivo del proyecto es analizar los productos que permiten gestionar de forma abierta y eficiente los recursos computacionales de un Centro de Procesamiento de Datos. El proyecto pretende analizar los nuevos modelos de operación en los CPDs, basados en la virtualización de los recursos computacionales (disco, cpu, memoria) y la automatización de los mismos, con el objetivo de la posterior implantación de un sistema abierto de gestión de la infraestructura, enfocada a funcionar como un entorno de desarrollo.

En la sección 2, se describe más extensamente el objetivo del proyecto y los requisitos iniciales que debe cumplir la infraestructura.

Una vez recopilada la documentación necesaria en la sección 3 se introducirán los fundamentos teóricos que permitirán diseñar la gestión de una infraestructura virtualizada según el modelo Cloud Computing. Definiremos los aspectos que caracterizan este tipo de infraestructuras en el apartado 4.

A continuación en el apartado 5, se realizará un estado del arte de las herramientas de gestión de entornos virtualizados que faciliten su integración y automatización. Centraremos nuestras conclusiones en los gestores que trabajen con licencias abiertas.

Se presentará una planificación detallada del proyecto, calculando así su coste asociado y planes de contingencia ante posibles riesgos (sección 6).

Analizando las herramientas de diseño, configuración y programación necesarias, así como las tecnologías utilizadas en la creación de los gestores (sección 7).

Finalizada la fase inicial de análisis, nos centraremos en aplicar la mejor metodología para la adopción de estas tecnologías en nuestro entorno. Se expone la arquitectura, diseño y implementación de cada uno de los gestores a estudiar (sección 8). La sección 9 se presentaran algunas de las pruebas realizadas con el entorno, con la finalidad de mostrar alguna de las funcionalidades implementadas.

Finalmente, la sección 10 presentará las conclusiones y metas alcanzadas tras la consecución del proyecto y marcará las futuras líneas de continuación y mejora.

2. Objetivos del proyecto

El objetivo del proyecto es analizar los diferentes gestores de entornos virtuales capaces de ofrecer para la infraestructura interna de *Atos Research and Innovation* un servicio similar al ofrecido por el proveedor Amazon EC2 de forma externa (IaaS).

Se realizará el diseño y implementación basándonos en los requisitos de un entorno de desarrollo/test, con la finalidad de acotar nuestro entorno de pruebas. El cometido de este análisis es reaprovechar ordenadores anteriores al 2006, sin soporte adicional del hardware para las nuevas técnicas de virtualización, y que han quedado en desuso.

Se busca diseñar una infraestructura enfocada a adaptarse a las condiciones específicas de los proyectos de este departamento de desarrollo.

Se trata de un entorno cambiante, encontramos un gran número de proyectos en los que el entorno de programación presenta una morfología y unos requisitos software muy dispersos entre si, incluso en ocasiones incompatibles. Otro de los factores a destacar, es que la duración de estos proyectos y la necesidad de recursos donde poder desarrollar es altamente variable, encontramos entornos en los que necesitamos los recursos durante meses, eventualmente durante algunas horas o puntualmente para realizar alguna demostración de las funcionalidades del software o de los prototipos implementados.

La virtualización de servidores puede encapsular cada proyecto, creando un entorno de trabajo propio, configurable según las necesidades específicas del programador. Un entorno virtualizado, extiende los beneficios de la virtualización sobre un conjunto de recursos físicos. Esto permite asignar que recurso virtual se ejecuta sobre cada recurso físico y simplifica la migración de estas máquinas entre los recursos de la infraestructura. Una gestión eficiente, nos permite afrontar la variabilidad de utilización de los recursos.

2.1. Requisitos del proyecto

- Profundizar en el estado del arte de los gestores de recursos virtuales.
- Realizar la implementación mediante herramientas opensource.
- Gestión del sistema centralizada y manejada por un administrador de la infraestructura.
- Re-utilización de ordenadores que han quedado en desuso.
- Entorno capaz de simular diferentes topologías de red y entornos de trabajo.

2.2. Rol del desarrollador de software

Las funciones del desarrollador son estrictamente las de configurar el entorno de programación y la de programar. No desea realizar funciones de instalación o configuración de equipos. Sus preocupaciones se basan en tener acceso sobre los recursos donde tienen que trabajar y que las características de estos recursos se adecuen a sus necesidades (SO, software instalado, espacio de disco, capacidad de computo, configuración de red, memoria).

2.3. Rol del administrador de sistemas

Tener la capacidad de ofrecer un entorno de desarrollo para los programadores de una forma simple y centralizada. Desde este mismo punto desea ser capaz de monitorizar el estado de estos recursos, ofreciendo la posibilidad de modificar sus características en caso de necesidad.

3. Fundamentos teóricos

A lo largo de esta sección presentaremos los fundamentos teóricos relevantes para la consecución de nuestro proyecto. Tras la recopilación de documentación, observamos el emergente interés de la comunidad emprendedora y científica por la adopción del modelo Cloud Computing, como tecnología que habilita un mayor control y dinamismo en los centros de procesamiento de datos. Este modelo se apoya en la virtualización para entregar recursos computacionales como un servicio (IaaS). La virtualización ejerce como tecnología clave en estos entornos dinámicos, ofreciendo una mayor flexibilidad a la hora de tratar los recursos físicos, pudiendo definir su ratio de utilización en cada máquina virtual.

3.1. El modelo Cloud Computing

3.1.1. Contextualización

Cloud Computing hace referencia a un fenómeno tecnológico emergente en los últimos años, se presenta como la evolución de la forma en que actualmente utilizamos y concebimos Internet para visualizarla como una fuente de servicios heterogéneos.

La perspectiva de proveer los recursos computacionales de forma similar a servicios habituales en la sociedad moderna, como el agua, el gas, la electricidad o las comunicaciones, fue introducido alrededor de 1955 [2] con el apelativo de *Utility Computing*, este concepto contempla la posibilidad de aprovisionarse de recursos computacionales, como procesamiento o almacenamiento, según las específicas necesidades del usuario final de estos recursos. No obstante, *Utility Computing* permanece como un concepto durante varios años, aletargado a la espera de su viabilidad tecnológica, en la década de los sesenta se siembra el campo de cultivo donde esta visión podrá ser factible, *Internet*. Es durante este periodo donde aparece la primera referencia futurista a ubicar la capacidad de cómputo en la red de manos de *John Gage* (Sun Microsystems) con la expresión *"The NetWork is the Computer 1984* [3] en un momento en que el ancho de banda no era suficiente.

Con la aparición del *WorldWideWeb* (1991), la evolución, globalización y profesionalización experimentada por Internet durante finales de siglo XX y principios del XXI ha hecho posible un ancho de banda capaz de afrontar estos retos, apoyándose en la aparición de tecnologías complementarias como son los servicios web *(WS)*, las arquitectura orientadas a servicio *(SOA)*, los sistemas distribuidos, el software as a service, el grid computing, la Web 2.0 o la virtualización.

3.1.2. Definición del modelo Cloud Computing

El paradigma del Cloud Computing cambia la ubicación de la infraestructura a la red con el fin de reducir costes asociados a la gestión de recursos hardware y software, esta migración exige un cambio de mentalidad y requiere una consolidada base de conocimientos.

La variedad de tecnologías en un Cloud hace que la visión general sea equivoca, se han postulado varias definiciones de *Cloud Computing* y sus funcionalidades:

Forrester ofrece esta visión [4]:

> *Cloud computing es una forma de estandarización de los entornos de TI basada en la capacidad - tanto de los servicios basados en Internet, software, o infraestructuras de TI - de ser ofrecidos por un proveedor de servicios que es accesible a través de los protocolos de Internet, desde cualquier ordenador, siempre disponible y escalable automáticamente para adaptarse a la demanda, ya sea a mediante el pago por uso o subscripción a estos servicios, dispone de una Web o interface de control programática permitiendo al cliente pleno control de auto-servicio.*

Gartner define el Cloud como [5]:

> *Cloud computing es un estilo de computación, escalable donde las capacidades de entornos TI son entregados como un servicio a clientes externos utilizando las tecnologías de Internet.*

Telefónica I+D lo define en 2008 como [6]:

> *Los Clouds en definitiva son un gran número de recursos generalmente virtuales fácilmente utilizables y accesibles tales como hardware, plataformas de desarrollo y/o servicios. Estos recursos pueden ser reconfigurados dinámicamente para adaptarse a una carga variable (escalable) así como permitir una óptima utilización de los recursos. Este conjunto de recursos puede ser explotado por el proveedor de la infraestructura a través de un modelo de pago por uso por medio de SLAs personalizados.*

Actualmente, no encontramos una definición estándar para el modelo, son tantas las diferentes definiciones al respecto del término Cloud Computing que existen documentos y artículos cuyo objetivo es recopilar todas estas definiciones[7][8][9].

A pesar de esto, sí es posible definir las características deseables que deben cumplir este tipo de infraestructuras (sección 4), considerándolo tanto desde el punto de vista de los usuarios de estos recursos como desde la perspectiva del proveedor de la infraestructura.

- *Perspectiva de usuario:*
  Los recursos son presentados a los usuarios de forma transparente con independencia de cuál sea su ubicación en la infraestructura, se da la

posibilidad de abastecerse únicamente de los recursos estrictamente necesarios, la provisión puede ser controlada por el proveedor o autogestionada por el propio usuario de una forma sencilla y intuitiva. Así pues, el usuario puede beneficiarse de un modelo de pago por uso de estos servicios mediante un SLA personalizado y conociendo a priori el coste que supondrá su utilización durante un cierto periodo de tiempo.

- *Perspectiva del proveedor:*
  Existen un conjunto de requisitos deseados en una infraestructura Cloud, son diseñadas con el objetivo de ser escaladas añadiendo nuevos recursos computacionales en caso de un pico en la demanda o simplemente utilizando los recursos de un proveedor externo. Se trata de infraestructuras flexibles, proporcionando una completa adaptación de sus recursos computacionales, como memoria, capacidad de procesamiento o almacenamiento.
  Permite desacoplar los recursos, favoreciendo el multipropósito de la infraestructura pudiendo compartir varios cometidos sin necesidad de preocuparse de como otros recursos han sido diseñados y sin comprometer su seguridad y privacidad. Las aplicaciones pueden ser compuestas utilizando y reutilizando servicios débilmente acoplados e independientes entre si, ofreciendo sus funcionalidades como un servicio para el usuario final en ocasiones disponible bajo demanda.
  Ayudándose de una estricta monitorización del comportamiento de la infraestructura como de sus recursos físicos, se pretende conseguir eficiencia y predictibilidad con el objetivo de crear una infraestructura auto-reparable en caso de fallo.

3.1.3. Niveles en el Cloud Computing

Cloud Computing se refiere tanto a la demanda de los servicios prestados a través de Internet, como al hardware y software en los centros de datos que permitan la prestación de estos servicios, esta visión es englobada por el concepto Everything as a Service (XaaS) [10].

Con el objetivo final de obtener una base de comprensión y favorecer una rápida adopción de este modelo, se utilizará una clasificación ya extendida. Esta consta de la disección del modelo Cloud en diferentes capas, cada capa proporciona una mayor nivel de abstracción en el servicio y cada una de ellas puede estar compuesta utilizando la capa de nivel inferior. Esta clasificación, en definitiva, es de gran utilidad para diferenciar los niveles de servicio en un Cloud y considerar las posibles inter-relaciones entre cada uno de ellos.

A continuación describiremos las 3 capas principales **SaaS, PaaS e IaaS** [figura 2], posteriormente **centraremos la atención en la capa de IaaS** (sección 8), describiendo el estado del arte (sección 5) en que se encuentran

los gestores open source capaces de proporcionar recursos de computación, red o almacenamiento bajo demanda.

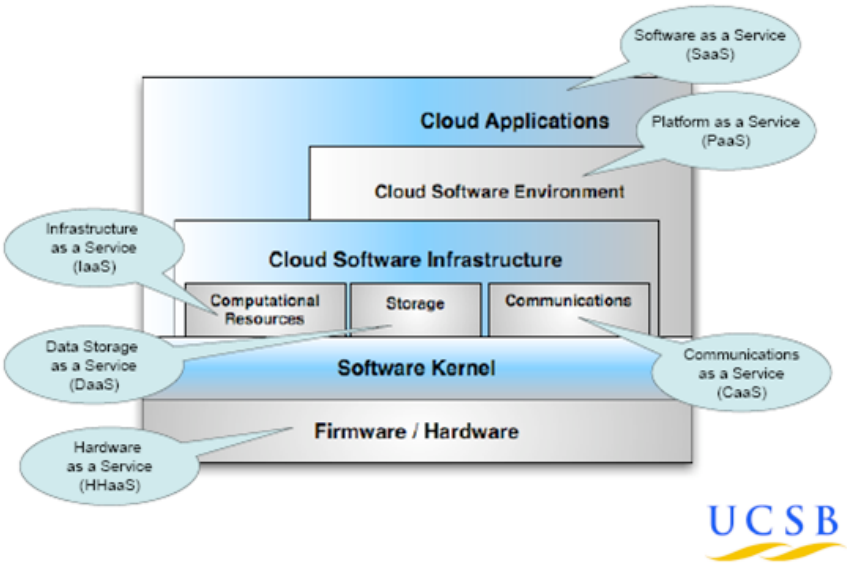


Figura 2: Lamia Youseff refleja en 5 niveles el modelo Cloud Computing

- *Software como Servicio (SaaS):* En la parte más alta, encontramos la capa de aplicaciones provistas en el cloud, es la primera que aparece incluso antes que el concepto Cloud Computing, también es el más extendido en la actualidad. Es un modelo de desarrollo de software donde las aplicaciones están alojadas como servicios prestados a los usuarios a través de Internet y disponibles bajo demanda.

  Un ejemplo destacado de software como servicio es Salesforce.com [11].

- *Plataforma como Servicio (PaaS):* En este nivel se encuentran los proveedores de entornos Cloud, suministran a los desarrolladores una interfície programática con un conjunto de APIs y entornos de ejecución bien definidos para facilitar la interacción y creación de los diferentes entornos y aplicaciones en el Cloud.
  En esta categoría se incluyen plataformas como Googles App Engine [12] o Microsoft Azure [13].

- *Infraestructura como Servicio (IaaS):* En ocasiones también llamado (Hardware como Servicio), plantea la posibilidad de aprovisionar parte de una infraestructura como un servicio, generalmente estas infraestructuras se presentan como un entorno virtualizado, aunque no necesariamente. Enfocado a evitar la infra-utilización de las infraestructuras, dando la posibilidad de reaprovechar los servidores, el espa-

cio en los centros de datos y los equipamientos de red.
IaaS esta ejemplificado como mayor exponente por Amazon Web Services Elastic Compute Cloud (EC2) [14].

Dentro de la tercera capa (IaaS) podemos diferenciar dos tipos de infraestructura ya sea pública (**Public/External Cloud**) o privada (**Private/Internal Cloud**).

Cuando hablamos de Internal Cloud nos referimos a un entorno de computación dentro de los límites de una organización y generalmente de su uso exclusivo (Private Cloud). Por el contrario cuando hablamos de External cloud este entorno computacional se encuentra fuera de los límites de la organicación (External Cloud), todo y que no es necesariamente accesible para el público en general (Public Cloud), sino que puede estar concevido para hacer su infraestructura disponible para alguna organización específica.

La fusión de estas dos visiones aún nos permites realizar una nueva subcategoria, los **Hybrid Clouds** entornos computacionales surgidos de la combinacion de ambas perspectivas intena y externa. Pueden presentarse como un entorno híbrido de forma continuada o realizar un depliegue dinámico de los procesos que se ejecutan internamente en una organización y extenderse a un public cloud en momentos de un pico en la demanda (cloudburst).

3.1.4. Aplicaciones (Business Models)

Siguiendo con la clasificación establecida en la sección 3.1.3, encontramos en el mercado una amplia gama de proveedores para cada uno de los niveles en el cloud. La figura 3 muestra algunos de los más relevantes.

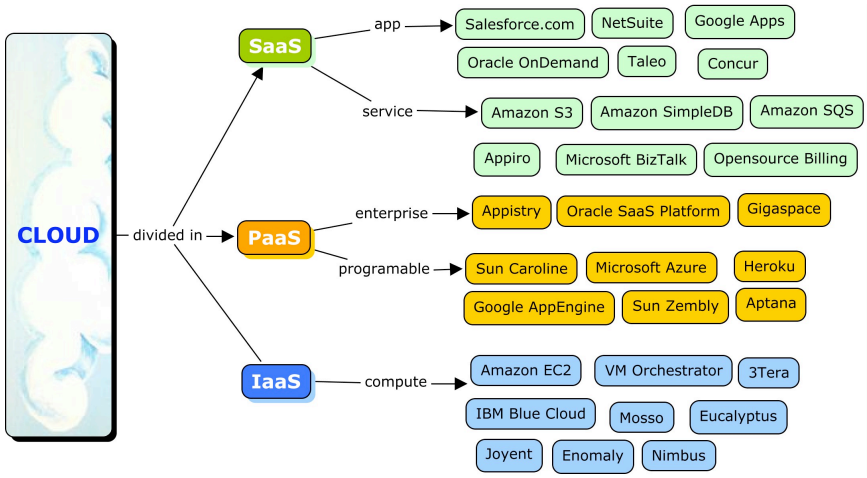


Figura 3: Diferentes proveedores en el cloud

Uno de los casos mas significativos es el de Amazon Web Services, este proeevedor de IaaS revoluciono el mercado ofreciendo la posibilidad a sus

usuarios de ejecutar servicios de computación de forma remota. Amazon EC2 ofrece maquinas virtuales Xen que utilizan los propios recursos de la empresa Amazon. Estas maquinas virtuales se presentan a los clientes de una forma simple y se delega en los usuarios la utilización de esos recursos. Complementario a este servicio ofrece otros servicios de infraestructura, como el almacenamiento , mediante el Simple Storage Service (S3) o red con el servicio Elastic IP entre otros.

El modelo de negocio es otro de los aspectos que lo caracteriza, Amazon EC2 se basa en un modelo de pago por uso. Disponemos de tres tipos de instancia según memoria, almacenamiento, procesamiento: Small (1.7Gb mem, 160GB,1core), Large (7.5Gb, 850GB, 2 cores) y Extra-Large (15Gb, 1690GB, 4 cores), la facturación se realiza dependiendo del tiempo de uso, el tipo de instancia y el consumo de red.

Es conveniente aclarar que no todos los procesos de negocio están igual de preparados para adoptar este modelo (figura 4). El hecho de tratar con información confidencial, susceptible de ser perdida o ser mal utilizada y que esta información sea almacenada fuera de las propias empresas y gestionada por terceros, es uno de los principales factores para que algunas empresas sean reticentes a utilizar proveedores de recursos externos que sigan este modelo.

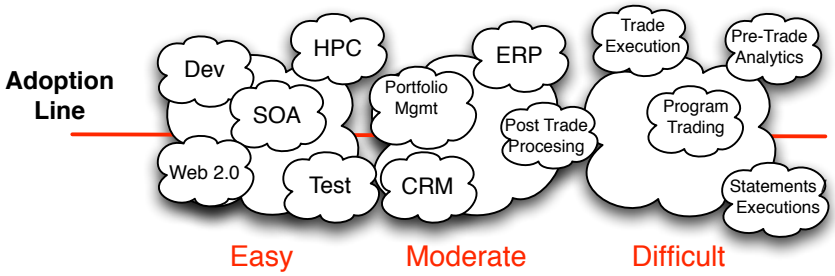


Figura 4: Areas de adopción [15]

El modelo Cloud Computing es una tecnología disruptiva, no obstante, las empresas que quieren adaptar sus infraestructuras pueden hacerlo de manera incremental, ya que trabajar en el cloud no implica que todos los recursos de una compañia estén en el cloud, no es necesario remplazar los servidores existentes, ni cambiar la organización o los procesos de los departamentos de TI.

3.1.5. Retos que plantea el nuevo modelo a la comunidad científica

Esta sección describe algunos de los temas de actualidad bajo investigación en el marco del Cloud Computing y la virtualización de servidores.

- Definición canónica del modelo Cloud.
- Validez de los casos de uso frente a las capacidades del modelo Cloud.
- Estándares tecnológicos en el Cloud Comuting.
- Interoperabilidad entre hipervisores, entre clouds.
- Meta-planifiación de máquinas virtuales en base a su rendimiento o al de la infraestrucstura.
- Capacidad de autonomía (self-*), incluye lo relativo a auto-gestión, auto-monitorización, auto-escalado, auto-reparación, etc.
- Elasticidad en base a la calidad del servicio. (QoS o QoE).
- Seguridad e integridad de datos en el Cloud. Cuestiones legales y problemas de confianza.
- Utilización de las transferencias de red y de protocolos de red que permiten mejorar el rendimiento de estos entornos.

Proyectos de invastigación

- **RESERVOIR:** *Resources and Services Virtualization without Barries[16]* es el proyecto Europeo (FP7) más activo en la investigación de infraestructuras virtulizadas y el Cloud Computing. Se inicio el febrero de 2008 y tiene planificada su duración hasta Enero de 2011, con el propósito de proporcionar una infraestructura, que soporte la puesta en marcha y despliegue de servicios bajo demanda y asegure la calidad de estos servicios.

  OpenNebula, uno de los gestores candidatos a implantar en nuestro entornos de pruebas, esta siendo parcialmente desarrollado en el marco de este proyecto.

- **NUBA:** Iniciativa nacional en la tecnologías Cloud dentro del plan AVANZA, *Normalized Usage of Business-oriented Architectures* (NUBA) tiene como objetivo facilitar el despliegue de servicios empresariales mediante el desarrollo de una plataforma Cloud federada. La investigación se centra en obtener interoperabilidad de tecnologías de virtualización y Clouds.

Iniciativas de estandarización

Las iniciativas de estandarización en el Cloud computing, se enfocan en proporcionar interoperabiliddad entre estos entornos. Una de las principales críticas por parte de los usuarios es la dificultad para cambiar de proveedor o interactuar con los servicios de otro proveedor.

- OGF Open Cloud Computing Interface (OCCI): Pretende especificar una API para la gestión de infraestructuras como un servicio (IaaS).
- Open Cloud Consortium (OOC): Consorcio formado por el Center for Computational Science Research (CCSR), enfocado al desarrollo de estándares y frameworks que ofrezcan interoperabilidad entre Clouds. Actualmente, gestiona un *testbed* sobre Cloud Computing (Open Cloud Testbed).
- SNIA Cloud Storage Interfaces: El objetivo SNIA Cloud es identificar los estándares para el almacenamiento de datos en el Cloud.
- Open Cloud Manifesto: Declaración que defiende un modelo abierto y basado en estándares, más de 200 organizaciones están suscritas.
- DMTF Open Virtualization Format(OVF): La especificación OVF es un formato abierto para la especificación de máquinas virtuales. El objetivo es la interoperabilidad entre las tecnologías de virtualización.
- Unified Cloud Interface Project: Interfaz Cloud abierta y estandarizada para la unión de varios proveedores Cloud.

3.2. Virtualización

3.2.1. Contextualización

La virtualización ha ejercido un impacto significativo en los entornos tecnológicos en los últimos años. Aún tratandose de una tecnología que establece sus inicios varias décadas atrás, actualmente la posibilidad tecnológica que ofrecen los nuevos hipervisores más eficientes, así como los beneficios para las organizaciones que esta tecnología implica, tales como reducción de costes, aprovechamiento y consolidación de la infraestructura, un mayor control sobre los recursos y simplificación en la gestión, hacen de la virtualización una de las tecnologías a tener en cuenta en la morfología que ofrecerán las infraestructuras en las organizaciones futuras.

La virtualización se acuño durante la década de los 60, donde entra en escena dos grandes proyectos computacionales M44/44X de IBM y Atlas, considerados parte importante del linaje evolutivo de la virtualización. IBM fue determinante en la evolución de esta tecnología, que permitía la partición lógica de grandes *mainframes* en distintas máquinas virtuales.
En la actualidad, las modernas redes y sistemas basados en la arquitectura X86, están sufriendo los mismos problemas de rigidez y subutilización del hardware que los *mainframes* sufrieron en la década de los 60. La amplia adopción de Windows y la aparición de Linux como servidor de sistemas operativos en la década de 1990 estableció los servidores X86 como un estándar de la industria.
VMware lanzo la virtualización para las plataformas X86 en la década de los 90 convirtiendose en uno de los referentes mundiales en este mercado, que durante finales de los 90 y principios del siglo XXI ha experimentado

una explosión de interés atrayendo a un gran número de soluciones y proveedores, que se han ido posicionando en este mercado, como por ejemplo, Microsoft, SUN y Xen.

3.2.2. Técnicas de virtualización

El cometido de las diferentes técnicas de virtualización es ejecutar múltiples máquinas virtuales sobre una máquina física. La tecnología encargada de realizar esta gestión se conoce como "Virtual Machine Monitor"(VMM)[17].

Podemos entender VMM como una porción de código encargada de gestionar múltiples instancias de sistemas operativos sobre un único sistema físico. Esta tecnología proporciona control sobre el procesador, la memoria y otros recursos, recolocándolos según los requerimientos de cada sistema operativo huésped. VMM proporciona los cimientos para llevar a cabo la gestión del entorno virtualizado; discos duros virtuales, políticas de automatización, gestión del ciclo de vida y recolocación de los recursos en tiempo real.

La arquitectura de los procesadores X86 no contemplaba las necesidades de acceso a los recursos del sistema por parte del VMM.

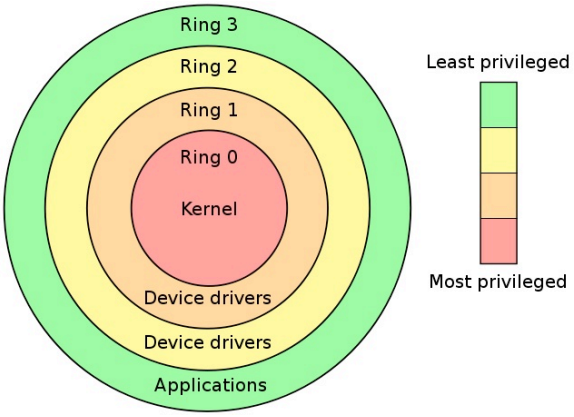


Figura 5: Anillos de privilegio para la arquitectura X86

Los sistemas operativos proporcionan diferentes niveles de acceso a los recursos, se conoce a estos niveles como anillos de protección. En las arquitecturas X86 tradicionales, los núcleos de los sistemas operativos ejecutan los accesos directos a la CPU desde el anillo 0, nivel que goza de mayores privilegios.

El problema surge cuando necesitamos ejecutar instrucciones en las arquitecturas X86 desde el anillo 0, que es el nivel que accede de forma directa al hardware del sistema. Con la virtualización, los sistemas operativos huésped no pueden ejecutarse en el anillo 0 ya que la VMM reside a este nivel.

A continuación, veremos como los diferentes técnicas de virtualización afrontan el problema para estas instrucciones que necesitan privilegios de ejecución en el anillo 0.

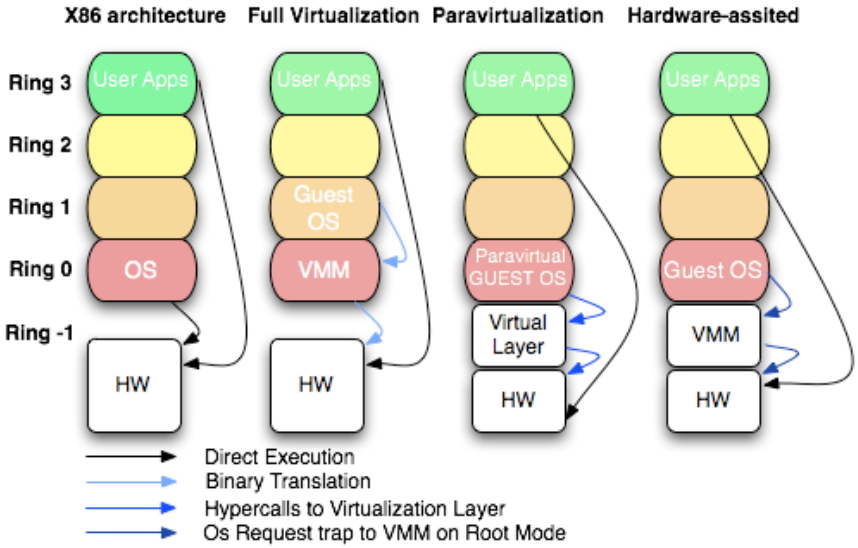


Figura 6: Técnicas de virtualización

Virtualización Completa

Es una técnica de virtualización que proporciona una completa simulación del hardware subyacente, Full-Virtualization se diferencia de las otras técnicas siendo la única que no requiere asistencia del hardware o del sistema operativo. El sistema huésped no tiene conocimiento que se esta virtualizandando y no requiere modificación.
Esto se consigue traduciendo el código del kernel reemplazando las instrucciones no-virtualizables por instrucciones que tienen efecto sobre el hardware virtual. A nivel de usuario el código se ejecuta directamente sobre el procesador para un mayor rendimiento.
El hypervisor traduce todas las instrucciones del sistema operativo sobre la marcha y guarda los resultados intermedios para usos futuros, en cambio, a nivel de usuario se ejecutan las instrucciones a velocidad nativa.
Algunos de los ejemplos de esta técnica de virtualización son VMWare Server, Parallels Workstation, VirtualBox o Microsoft Virtual Server.

Paravirtualización

A diferencia del caso anterior, la paravirtualización conlleva modificar el núcleo (kernel) del sistema operativo con el fin de reemplazar las instrucciones no-virtualizables. Mediante hypercalls se permite comunicar de forma

directa con la capa de virtualización o hipervisor. El hipervisor a su vez, proporciona una interface capaz de realizar operaciones críticas en el kernel, tales como, gestión de la memoria, manejo de las interrupciones o control del tiempo. A esta modificación se la conoce como "portabilidad", esta técnica a priori ofrece ventajas en el rendimiento apoyándose en la capa de virtualización, no obstante esta apreciación puede variar enormemente dependiendo de la carga de trabajo a la que está sometida.
El proyecto de código abierto Xen es un ejemplo de paravirtualización que virtualiza el procesador y la memoria utilizando una versión modificada del núcleo de Linux.

Virtualización asistida por hardware

Los fabricantes de hardware, conscientes de la problemática existente entre las técnicas de vitualización y el acceso al sistema de hardware, han desarrollado nueva funcionalidades que permiten simplificar esta interacción. Intel y AMD realizaron cambios a sus arquitecturas X86 facilitando así la función del VMM (Intel-VT, AMD-V). Esta nueva técnica introduce un nuevo nivel de privilegio por debajo del anillo 0. El hipervisor puede gracias a estas nuevas funcionalidades, ejecutarse en el anillo -1, permitiendo al sistema operativo huésped ejecutarse en el anillo 0, y acceder de forma directa a los recursos sin necesidad de emulación o modificación del sistema operativo.

Algunos de los ejemplos de esta técnica de virtualización son WMWare, Xen HVM, KVM, VirtualBox.

4. Características del modelo

4.1. Auto-servicio

Se simplifica la gestión de los administradores de la infraestructura. Generalmente por medio de una interfaz web o mediante linea de comandos, los usuario son capaces de aprovisionarse de los recursos que necesitan de una forma sencilla.

El usurario final no necesita conocer como la infraestructura ha sido diseñada, conocer cuales son las tecnologías subyacentes, ni aprender una extensa documentación para hacerla funcionar.

4.2. Escalabilidad

El concepto de escalabilidad se aplica al rendimiento de la infraestructura y la capacidad de esta de dar soporte a las necesidades de los usuarios.

4.3. Elasticidad

La elasticidad o flexibilidad se refiere a la capacidad de la infraestructura a escalar en ambas direcciones, tanto para dar soporte a las necesidades de miles de usuarios, como para dar servicio a un único usuario.

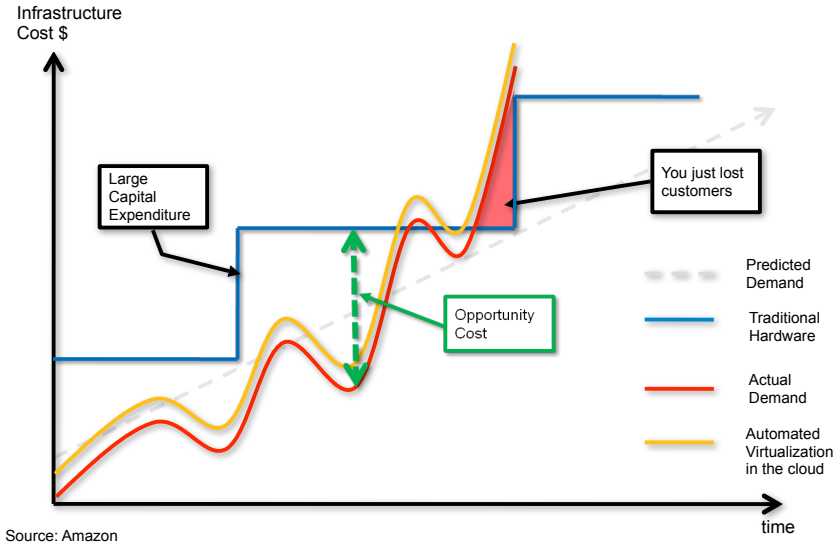


Figura 7: Cloud Computing gestiona la capacidad de picos y valles sin incurrir en gastos CAPEX

4.4. Multipropósito

El sistema puede estar diseñado para que varios usuarios puedan compartir la infraestructura y le den un uso específico, sin comprometer la privacidad y seguridad de los datos de cada usuario.

4.5. Orientada a servicio

Las aplicaciones pueden ser compuesta utilizando servicios débilmente acoplados que conjuntamente ofrecen un servicio final. El sistema debe tolerar la posible caída de un servicio, haciendo que esto no afecte a los demás.

4.6. Disponible bajo demanda

Los recursos pueden ser entregados bajo demanda en un corto plazo o reservados con antelación.

La gestión del hardware se abstrae del consumidor, estos consumidores no incurren en gastos de infraestructura ni el los costes asociados a la gestión de la misma. Esto permite eliminar los costes iniciales de adquisición de la infraestructura (CAPEX) y pasar directamente a consumir bajo demanda acortando el tiempo de acceso al mercado. El consumidor únicamente incurre en gastos de operacionales (OPEX), como gastos de explotación o funcionamiento.

4.7. Eficiencia / Predictibilidad

Mediante una estricta monitorización de los recursos de la infraestructura se permite planificar y predecir su comportamiento. Se puede tener control sobre los recursos que están siendo utilizados y el estado de los servicios de que disponen los usuarios. Conocer el uso que se le da a la infraestructura permite avanzarse a los acontecimientos actuando en consecuencia, esto es muy importante en un entorno virtualizado donde varios ordenadores trabajan simultáneamente, ya que se añade complejidad a la gestión de esta infraestructura ahora controlada por software.

4.8. Autorreparable

La probabilidad de un fallo software es más alta que la de fallada de hardware, no obstante asumiendo que los fallos ocurren y son inevitables, la recuperación del sistema software se realiza de una manera mucho más rápida, asegurando la continuidad de negocio, que lo que supondría la recuperación de un sistema tradicional, compra de nuevo recurso hardware, instalación, configuración y test. Las máquinas virtuales se comportan de manera flexible pudiendo desplegarse sobre otros recursos hardware, así pues

se establece un tiempo de perdida de servicio, como máximo el tiempo que tarda una máquina virtual en volver a ser desplegada (el tiempo de reinicio).

Existen dos formas de trabajar con VM. La primera, de forma persistente, almacenando a cada cambio la información para no ser perdida. La segunda, almacena la información cuando se le indica, en caso de fallo en la máquina se puede recuperar el sistema desde el ultimo backup.

Se necesita un sistema que realice snapshots consistentes de las instancias en funcionamiento de las máquinas virtuales, generalmente se crea un archivo comprimido que incluye toda la información almacenada en la máquina virtual, esto incluye los archivos de configuración.

Hay diversas maneras de abordar este reto para tratar de dar coherencia y consistencia a nuestro entorno virtualizado.

- Parar completamente durante la copia de seguridad la máquina virtual que se quiere almacenar, este método conlleva un gran periodo de inactividad para la máquina virtual.
- Suspender la ejecución de la máquina virtual durante el backup, a su vez se sincronizan (rsync) los archivos que han podido ser modificados desde el anterior punto de backup. El tiempo de inactividad es mínimo ya que una vez sincronizados los ficheros la máquina puede reprender su trabajo.
- No parar la máquina virtual y realizar el backup sin tiempo de inactividad (LVM2), los proveedores de hypervisores ha realizado grandes avances en este sentido, consiguiendo realizar tanto backups como migraciones en tiempo real.

4.9. Acuerdo de Nivel de Servicio (SLA)

Los servicios se prestan bajo un contrato entre las partes, la negociación de los términos es siempre una difícil tarea y debe ir acompañada de indicadores que permitan evaluar la prestación del servicio.

Se esta trabajando en la prestación de SLAs de forma dinámica capaces de cuantificar el uso que hacemos del servicio.

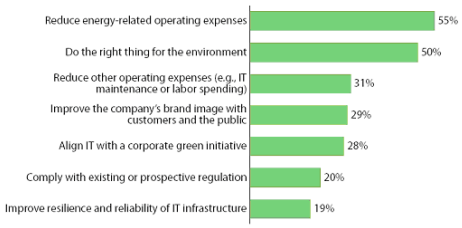
4.10. Modelo de facturación

Es quizás uno de los aspectos que han producido un mayor cambio, la posibilidad de realizar un venta al detalle de los recursos hardware que el usuario final quiere disponer. A priori los usuarios conocen las tarifas derivadas de la utilización de estos recursos, pagando únicamente por lo que reciben. Esto permite planificar los costes asociados a los nuevos despliegues.

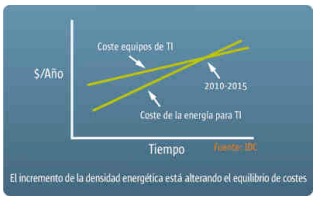
El pago por uso de los recursos puede ser facturado por el consumo de recursos en una hora, mensualmente, anualmente, según las características de la imagen, la cantidad de almacenamiento requerida o del tráfico de red.

4.11. Sostenibilidad

El paradigma Cloud como la virtualización se presentan como herramientas decisivas para reducir la emisiones de CO2 y el consumo energético en los CPDs.



(a) Factores de negocio para la adopción de soluciones de eficiencia energética en las infraestructuras TIC



(b) Comparativa costes energía frente a costes de equipamiento

Figura 8: Eficiencia energética.

Las TIC se posicionan como uno de los principales sectores consumidores de energía a nivel mundial, según Gartner [18] este sector es el responsable del 2 % de las emisiones de CO2 a nivel mundial. De este 2 % uno de los elementos más significativos son las emisiones producidas por el Centro de Procesamiento de Datos (CPD) [19].

Una posible solución para mitigar estas emisiones es la renovación del equipamiento en los CPDs por otros con mejor ratio rendimiento-consumo, aunque se trata de una solución a corto plazo y con un gran coste derivado. La solución que aporta el modelo Cloud computing conjuntamente con la virtualización es mejorar la gestión de la infraestructura permitiendo optimizar y planificar el uso.

5. Estado del arte de los gestores de máquinas virtuales

Como obligada consecuencia del crecimiento de la popularidad de los términos como virtualización, utility computing[2], everything as a service [3] o cloud computing [4] un número considerable de productos comerciales y proyectos de investigación se han ido desarrollando en la línea de superponer de forma dinámica las máquinas virtuales sobre recursos físicos, extendiendo los beneficios de la virtualización sobre un conjunto de recursos.

5.1. Amazon EC2

No se trata de un gestor sino de un Cloud Público, Amazon EC2 es el caso que ejemplifica mejor la solución que se quiere adoptar, aunque en la visión externa a una organización. Es un servicio web ofrecido remotamente que proporciona capacidad de cómputo en el cloud (Internet), redimensionable bajo demanda. Es el principal caso de éxito en las infraestructuras como servicio (IaaS), proporcionando tanto a particulares como organizaciones capacidad de cómputo y almacenamiento, así como un sistema de pago por uso de los recursos.

5.2. Gestores comerciales

Aunque el foco del proyecto es analizar los gestores *open-source*, se incluye alguno de los gestores comerciales capaces de realizar la gestión de una infraestructura virtualizada.

VMWare Infrastructure 3

Permite que múltiples sistemas operativos y sus aplicaciones se ejecuten en máquinas virtuales de manera independiente y sin realizar ninguna modificación a la vez que comparten recursos físicos. Este conjunto de herramientas ofrece capacidades completas de virtualización, administración, optimización de recursos, disponibilidad de aplicaciones y automatización operacional.

Platfom VM Orchestrator

Plataforma de auto-servicio de maquinas virtuales, permite a los usuarios de su portal web crear y gestionar sus propias maquinas virtuales. Platform VMO está fuertemente ligado a XenServer.

IBM Virtualization Manager

Permite visualizar y gestionar los recursos físicos y virtuales desde una única consola. Simplifica la gestión de los entornos virtualizados VMware, Xen, Microsoft® y POWER pudiendo trabajar conjuntamente con VMware VirtualCenter.

Microsoft System Center

Es la solución aportada por Microsoft para la gestión de los recursos físicos y virtuales sobre los centros de datos. Facilita la integración y automatización en las organizaciones TI. Su hipervisor por defecto es Hyper-V, creado por Microsoft, aunque también soporta el despliegue de máquinas VMWare.

5.3. Gestores no comerciales

Los siguientes gestores, son los candidatos a analizar sobre nuestro entorno de prueba. En la conclusión del estado del arte (sección 5.5), se hará hincapié en las tecnologías de virtualización soportadas, la madurez en la que se encuentran y la capacidad de interacción con un proveedor externo (EC2).

Eucalyptus

El objetivo del proyecto es promover la investigación y desarrollo de las tecnologías que ofrecen servicios elásticos, útiles y en la nube, así como el estudio de las estrategias de asignación de recursos, acuerdos a nivel de servicio (SLA), políticas y modelos de uso. Ha sido desarrollado en el laboratorio MAYHEM por el Departamento de Ciencia y Computación de la Universidad californiana de Santa Barbara, inicialmente como una herramienta para la investigación del cloud computing.

Soporta Xen y KVM como tecnologías de virtualización.

Nimbus

Se presenta como una de las herramientas open source más consolidadas en el mercado, permite convertir un clúster en una infraestructura cloud orientada a servicio. Está enfocado en desarrollar una solución de computo en la nube centrado en las iniciativas científicas, aunque su diseño soporta casos de uso fuera del marco científico.

Implementación basada en el hypervisor Xen, se espera que soporte KVM próximamente.

Enomaly

Fundado en noviembre de 2005 por Enomaly Inc., la plataforma de computación elástica, se centra en diseñar, desarrollar y gestionar las aplicaciones virtuales en la nube. Se simplifican los costes administrativos y las cargas de trabajo del sistema. Se centraliza la administración de la infraestructura desde una interface web permitiendo al personal de TI, un despliegue eficiente de sus planes de desarrollo, automatización, escalado y balanceo de carga de sus máquinas virtuales.

Soparta Xen, KVM y VMWare.

Open Nebula

Es un *middleware* que está siendo desarrollada por la Universidad Complutense de Madrid en el proyecto RESERVOIR, proyecto europeo de investigación más activo en virtualización de infraestructuras y cloud computing. ONE actúa como motor open source de una infraestructura virtual, permitiendo el despliegue dinámico y la recolocación de las máquinas virtuales sobre un conjunto de recursos físicos.

Soporta Xen, KVM.

Emotive

Emotive ha sido desarrollado por Barcelona Supercomputing Center, actualmente es una de las tecnologías utilizada en diferentes proyectos europeos como BREIN y SORMA.

Emotive permite a los usuarios una gestión flexible de las tareas en los entornos virtualizados. Esta compuesto por tres capas, el planificador que decide donde se ejecuta la tarea, el gestor de recursos virtuales y por último una infraestructura distribuida de datos.

Emotive soporta Xen como tecnología de virtualización.

Abicloud

Abicloud ha sido desarrollado por Abiquo un *start-up* Barcelonés que ha entrado con fuerza en el Cloud Computing ofreciendo sus servicios tanto en el mercado europeo como estadounidense. Abiquo llega al cloud computing a través de la investigación de los entornos grid computing.

Abicloud es una software abierto, para la creación y gestión de infraestructuras basadas en entornos heterogéneos. Las herramientas de que dispone ofrecen la posibilidad al usuario de escalar, monitorizar, automatizar y aprovisionarlos de servidores, almacenamiento y capacidad de red según sus necesidades.

La primera versión del producto(0.6) se lanzo con soporte para VirtualBox, actualmente soportan Xen, KVM y VMware próximamente y ofrecen

la posibilidad de configurar las VM mediante Open Virtualization Format (OVF).

5.4. Comparativa

En la tabla 1 se muestra la comparativa inicial realizada a los gestores *open-source* susceptibles de convertirse en candidatos a implementar. En esta valoración inicial centraremos la atención en las diferentes técnicas de virtualización soportadas por cada uno de los gestores, ya que los ordenadores donde se realizaran las pruebas de diseño no admiten virtualización asistida por hardware al ser modelos anteriores al 2006.

Gestores	VMM	SO	Monitorización	Licencia
Eucalyptus	Xen, KVM	Linux	Integrada CLI	FreeBSD
Enomaly	Xen, KVM, VMware	Win, Linux	Vía web	AGPL
Open Nebula	Xen, KVM	Linux	Itegrada CLI	General Public License.
Nimbus	Xen	Linux	Nagios or other third-party tools	Apache License 2.0.
Emotive	Xen	Linux	No	?
Abicloud	Xen, KVM, VirtualBox, OVF	Linux, Windows, Solaris	Integrada web	MPL 1.1 server components, CPAL 1.0 client

Cuadro 1: Tabla comparativa gestores no comerciales.

5.5. Conclusiones

Tras el estudio preliminar encontramos ciertos requisitos que nos permiten contrastar la elección de unos gestores en frente de los otros. La técnica de virtualización elegida para nuestra infraestructura es **Xen *open-source***, está técnica de paravirtualización es la que nos asegura un soporte consistente para nuestra infraestructura de pruebas (modelos 2006), así como la única que nos facilitará la interoperabilidad con el cloud público **Amazon EC2** (basado en Xen), los gestores que cumplen los requisitios, en la fecha fijada por la toma en decisión, son **Eucalyptus y OpenNebula**.

Descartamos abicloud al no haber realizado el lanzamiento de la versión estable de su producto para el hypervisor Xen, antes de la fecha fijada como medida de contingencia (sección 6.6). Salió al mercado a mediados de Abril

con soporte para VirtualBox, posteriormente ha ido liberando el soporte para otras tecnologías, algunas de ellas pendiente de prueba.

Enomaly todo y soportar Xen ofrece algunas de sus funcionalidades web únicamente para los hipervisores que soportan virtualización asistida por hardware. La comunidad del producto, no cuenta con documentación para los desarrolladores de la infraestructura, ni facilita la interacción con EC2. Lo que hace que descartemos Enomaly como gestor elegido. No obstante se realizaran pruebas del producto para un único ordenador virtualizado (anexo C).

Nimbus a priori parece uno de los gestores más competentes, ayudado por la extensa comunidad de desarrolladores que poseen sus productos (Globus Toolkit), aunque los requisitos de configuración de red serán difíciles de afrontar en nuestro entorno de pruebas. Las herramientas que ofrece Globus están enfocadas a los entornos Grid, lo que no se adecua a nuestro objetivo.

Emotive, producto lanzado por Barcelona Supercomputer Center es otro de los gestores que cumplen los requisitos de nuestra infraestructura, soportando Xen como hipervisor. Aunque no se encuentra en un estado de madurez suficiente para asegurar el despliegue de nuestra infraestructura hacia Amazon EC2.

Eucalyptus y Open Nebula son dos de los productos más activos en la investigación de este tipo de infraestructuras, ambos provienen del mundo universitario, americano y europeo respectivamente. Debido a este carácter no lucrativo, ofrecen de forma abierta sus funcionalidades, requisito indispensable en nuestro proyecto. Ofrecen soporte para Xen paravirtualizado y poseen herramientas que nos permite realizar la interacción con EC2.

6. Planificación del proyecto

En esta sección, se describe la planificación para llevar a cabo los diferentes objetivos planteados. Se desglosaran las tareas a realizar y se estimara en horas las dedicaciones que conlleva el cumplimiento de cada tarea. Finalmente se incluye un diagrama de la planificación y unas normas de actuación en caso de riesgo en la viabilidad del proyecto.

6.1. Tareas de gestión

- **Reuniones de seguimento (30 h):** Reuniones con los directores tanto de la empresa como de la universidad para realizar el seguimiento.
- **Formación (60 h):** Formación en virtualización, asistencia a eventos.

6.2. Tareas de documentación

Estimación: **(120 horas)**

- **Introducción al Cloud Computing (30 h):** Introducción al modelo, recopilación de documentación ya publicada, soluciones aportadas y retos abiertos.
- **Introducción a la virtualización(30 h):** Introducción a la virtualización, recopilación de documentación existente y técnicas adecuadas para el proyecto.
- **Estado del Arte Virtualización (30h):** Recogida de requisitos deseados en nuestra infraestructura e investigación sobre las soluciones existentes.
- **Estado del Arte gestores de la infraestructura (30 h):** Investigación sobre las soluciones existentes: comerciales y no comerciales: Funcionalidades ofrecidas por cada uno de ellos.

6.3. Tareas de diseño, configuración e implementación

Estimación: **(960 horas)**

- **Diseño del entorno de pruebas (30h):** Establecer la arquitectura a diseñar, identificar los componentes necesarios y decidir que tecnología de virtualización es la adecuada.
- **Entorno de programación (12h):** Escoger, instalar y configurar las herramientas necesarias para interactuar con la infraestructura y sistema de control de versiones. Establecer un SO base con el que trabajar.

- **Analisis de VMM (30h):** Escoger, instalar y configurar las herramientas necesarias para interactuar con el hipervisor.
- **Experimentación + Analisis Xen Open Source (30 h):** Diseño de los componentes básicos que intervienen en la configuración del hipiervisor y creación de VM.
- **Configuración Entorno de pruebas (30 h):** Configuración del entorno virtualizado, adecuado a los gestores con los que va a trabajar y a la morfología de la red.
- **Analisis de funcionalidades (30 h):** Establecer versiones estables con las que experimentar y describir funcionalidades a implementar.
- **Implementación Enomaly (30 h):** Configuración del entorno Enomlay.
- **Implementación Open Nebula (30 h):** Configuración del entorno Open Nebula.
- **Implementación Eucalyptus (30 h):** Configuración del entorno Eucalyptus.
- **Provisionamiento Open Nebula (30 h):** Gestión centralizada y aprovionamiento con One.
- **Provisionamiento Eucalyptus (30 h):** Gestión centralizada y aprovionamiento con Eucalyptus.
- **Open Nebula + Amazon EC2, S3 (30 h):** Provisionamiento desde One a EC2
- **Eucalyptus + Amazon EC2, S3 (30 h):** Provisionamiento desde Eucalyptus a EC2 ,S3
- **Paquetización imágenes Xen (30 horas):** Creación VM necearias para los ejemplos, y entornos de programación habituales en nuestro entorno de desarrollo.
- **Implementación ejemplo Amazon EC2, S3 (30 h):** Video de provisionamiento bajo demanda de Eucalyptus a EC2 ,S3 de un entorno LAMP.
- **Implmentación ejemplo Open Nebula(30 h):** Provisionamiento desde One a nuestra infraestructura de un entorno de desarrollo LAMP.
- **Implementación ejemplo Eucalyptus (30 h):** Video del aprovisionamiento de VM sobre nuestros nodos en Eucalyptus.

6.4. Tareas de redacción

Estimación: **(420 horas)**

- **Informe previo (60 h):** Preparación del informe previo a la aceptación del proyecto.
- **Redacción de la memoria (360 h):** Recopilación de todo el trabajo realizado y redacción de la memoria.

Captura de pantalla del Diagrama de Gantt para la planificación (ampliada en el anexo B.

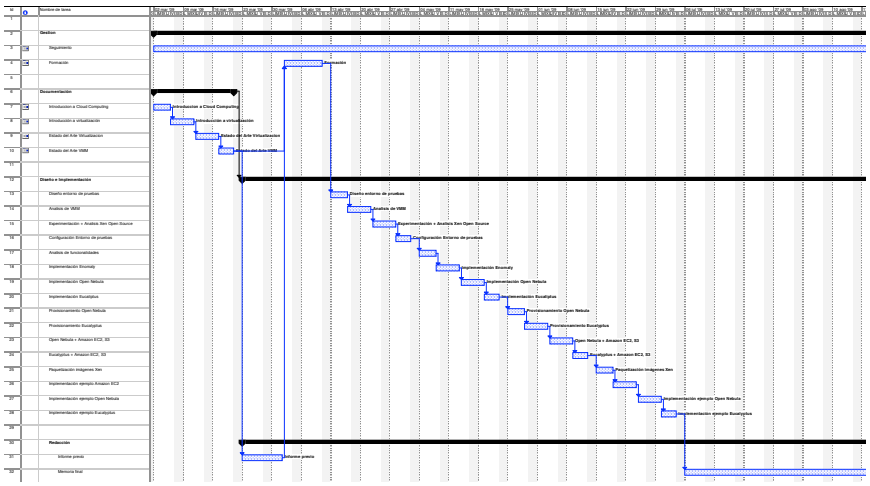


Figura 9: Diagrama Gantt

6.5. Coste asociado al proyecto

Los gastos del proyecto en su mayoría se deben a costes asociados a los diferentes perfiles del personal involucrado. Se han dividido estos recursos humanos según cuatro roles, el del director formado por el tutor del proyecto y tutores de la empresa. Un Analista funcional encargado de recopilar la información necesaria para llevar a cabo el diseño general de la infraestructura a implantar. Un Analista de sistemas y programador cuya misión es la configuración y puesta en marcha del entorno así como la implementación de cada uno de los gestores y despliegue de máquinas virtuales en nuestro entorno de pruebas.

A los recursos humanos hemos de añadir los gatos derivados de la infraestructura física en si misma, esta incluye 6 ordenadores de pruebas con su consecuente gasto en electricidad y refrigeración así como plataformas de pruebas y desarrollo. En el apartado de formación se engloba los eventos

y cursos realizados durante el periodo del proyecto. Por último se incluye un apartado de gastos generales como son servicios de transporte, servicio de reprografía, dietas. No es necesario incluir ningún importe en concepto de licencias, ya que unos de los requisitos del proyecto es ser desarrollarlo mediante herramientas abiertas.

Concepto	**Horas**	**Dotación**	**Total**
Analista sistemas	800	20 €/h	16 000 €
Analista funcional	200	25 €/h	5 000 €
Director	60h	32 €/h	1 920 €
Máquinas	-	(supeditado a incremento según número nodos)	800 €
Formación	-	400 €	300 €
Otros gastos	-	100 €	100 €
Coste total			24 120 €

Cuadro 2: Estimación coste asociado al proyecto

6.6. Riesgos asociados al proyecto

La tabla 3 muestra los diferentes riesgos considerados a priori y que se pueden ir presentando a medida que avanza el proyecto, la tabla tienen asociado a cada uno de los riesgos una acción de contingencia que permitirá eliminar o mitigar el riesgo.

Riesgo	**Acción contingencia**
El proyecto no alcanza resultados	Aún no habiendo alcanzado los objetivos, el proyecto proporciona unos conocimientos base sobre el modelo y una profundización del estado del arte inicial de los gestores, que permitirá definir cual o cuales son los gestores adecuados para un entorno de desarrollo para los modelos testeados en la infraestrucutra de ARI dentro de Atos Origin.
Ordenador no soportan virtualización asistida por hardware	Realizar las pruebas únicamente con la tecnologías de virtualización soportada en estos sistemas
Gestores en las primeras versiones estables y en continuo desarrollo	Realizar las pruebas con versiones estables de los gestores anteriores a Abril de 2009
Entorno de pruebas limitado y muchos gestores a estudiar	Descartar el estudio de gestores según interes de la empresa o debido a que no cumpla alguna de las acciones de contigencía anteriores

Cuadro 3: Riesgos/Acciones derivadas del proyecto

7. Entorno y lenguajes de programación

Esta sección refleja las diferentes tecnologías involucradas en nuestro entorno. Se enuncian las tecnologías utilizadas por Eucalyptus y Open Nebula, nuestros gestores implementados en el entorno de pruebas. Por otro lado, se repasan las tecnologías necesarias para la configuración del entorno, donde estos gestores están instalados.

7.1. Tecnologías en Open Nebula

- La construcción del sistema se realiza con **scons** una herramienta abierta para la construcción de software, similar al clásico *Make*.
- La estructura del almacenamiento se realiza mediante **sqlite**, una base de datos relacional contenida en una librería escrita en **C**.
- El núcleo del gestor esta escrito en **C++**.
- La mayoria de drivers están programados en **Ruby**.
- El driver Information Manager es el encargado de solicitar la información de los nodos remotos. Estas solicitudes pueden ser ejecutables o scripts en cualquier lenguaje o formato soportado por los nodos.
- El Transfer Manager utiliza **shell scripts** para describir cada unos de los comandos capaces de realizar acciones sobre las imágenes. Estos pueden ser programados en cualquier otro lenguaje.
- En la versión 1.4 el driver de WMware esta progamado en **JAVA**.
- La comunicación entre el núcleo y los drivers es un **protocolo de mensajes de texto (ASCII)** y utiliza pipes de unix entre los procesos del núcleo y los drivers.
- El protocolo de comunicación del cliente es **xmlrpc**, la librería utilizada es http://xmlrpc-c.sourceforge.net/ y utiliza un servidor empotrado **abyss**.
- La comunicación con los nodos donde se ejecutan las VM se realiza a través de **SSH**.

7.2. Tecnologías en Eucalyptus

- El Cloud Controller (CLC) es una programa en **JAVA** que ofrece un servicio web así como una iterfaz web que permiten interactuar con la infraestructura al mundo exterior.
- El Cluster Controller (CC) encargado de la planificación y el control de red a nivel de clúster está escrito en **C**.

- El Node Controller (NC), que realiza en control sobre el hypervisor que se esta ejecutando en los recursos físicos, al igual que el CC esta escrito en **C**. Y la interacción con los recursos se realiza mediante la invocación de scripts en **Perl**.
- El Walrus Storage Controller (WS3) implementa las APIs **Representational State Transfer-REST** y **Simple Access Object Protocol-SOAP**.
- La gestión de imágenes se realiza a través de las herramientas de Amazon (ec2-api-tools) escritas en **JAVA**.
- El almacenamiento de imágenes, es realizado al igual que el caso de la gestión por herramientas provistas por Amazon (ec2-ami-tools), estas están escritas en **Ruby**.

7.3. Entorno de pruebas

La pruebas se realizan en una subred (**DMZ**) de área local (LAN) tanto para evitar problemas de seguridad con la red interna de AtosResearch & Innovation, como para trabajar de forma independiente con el exterior. El escenario de pruebas esta formado por **dos ordenadores** que ejercen como máster de la infraestructura, donde están instalados los gestores a probar. Cada gestor controla **dos nodos** cada uno de diferente modelo, ambos modelos anteriores al 2006.

El sistema operativo elegido para las configuraciones es **Centos 5.2**, la elección de esta distribución se basa en los conocimientos del departamento de desarrollo y al ser una de las distribuciones referenciadas por los gestores en su documentación.

Ambos entornos disponen en los nodos del paquete **xen-tools** escrito en **Perl**, para la creación de nuevos dominios en Xen. Y **libvirt** una API para interactuar con las funcionalidades virtuales sobre Linux, escrita en **C**, aunque dispone de *bindigs* para otros leguages (C++, Python, Perl, Ruby, Java, C Sharp).

7.3.1. Entorno OpenNebula

- Shell Scripts para interactuar con el gestor.

7.3.2. Entorno Eucalytus

- EC2-API-TOOLS, escrito en JAVA.
- EC2-AMI-TOOLS, escrito en Ruby

8. Cloud interno

Esta sección aborda el núcleo del proyecto, después de hacer un repaso a los fundamentos teóricos del modelo (sección 3) y realizar una comparativa inicial de los gestores de infraestructura (sección 5). Se desea adoptar este modelo para un **entorno de desarrollo/testing** que será utilizado por los desarrolladores de la propia empresa. La implantación se debe realizar de forma incremental para no afectar a los procesos de trabajo ya existentes. A continuación se detallan los argumentos que llevan a considerar este tipo de modelo para esta clase de entorno.

Beneficios que aporta el modelo Cloud Computing a nuestro entorno:

- **Multiproposito:** Los diferentes proyectos conllevan unos requisitos específicos de software, mediante la vitualización se consigue desacoplar estos requisitos pudiendo trabajar cada desarrollador en un entorno programático específico para su proyecto.
- **Ratio de utilización:** Antes de la llegada de la virtualización el aprovechamiento de los recursos estaban por debajo del 50 % (10-15 % eran los ratios habituales)[20], gracias a esta técnica se permite encapsular las cargas de trabajo y transferirlas a los sistemas menos sobrecargados.
- **Ahorro espacio físico:** La expansión de servidores es otro de los asuntos a considerar, la virtualización permite consolidar diversos sistemas virtuales mitigando la expansión en unos pocos sistemas físicos. Cuando esta expansión sea justificada por causas de negocio podemos decidir si proveernos de nuevos recursos físicos o contratar estos recursos a terceros.
- **Ahorro energético:** La electricidad necesaria en los entornos computacionales no es ni mucho menos un factor despreciable, cualquier posible re-aprovechamiento de la infraestructura y mejora del ratio de utlización conlleva consigo una disminución de las emisiones de CO2. La capacidad de estos sistemas de ser elásticos nos permite incluso detener los recursos que están siendo infrautilizados, dejando así de consumir energía.
- **Continuidad de negocio:** Se proporcionan nuevas técnicas de recuperación para este tipo de entornos, al abstraernos completamente del hardware subyacente en caso de desastre se podrá recuperar nuestro entorno de trabajo sobre otro hardware en la empresa o simplemente contratando eventualmente esos recursos a proveedores externos bajo demanda. Nuestro entorno no implementa técnicas de auto-reparación, no obstante un fallo no provoca la perdida de los archivos fuentes de la máquina.

- **Reduce costes de operación:** Se reducen las cargas de trabajo administrativas, centralizando la gestión de la infraestructura. Estos entornos permiten preconfigurar servidores virtuales con características ampliamente demandadas (entornos de programación, servidor LAMP, etc), mediante un repositorio se puede controlar los tipos de instancia que se esta utilizando así como todos los detalles de su configuración incial (cpu, memoria, disco, SO,usuarios, programas instalados).

- **Privacidad:** Permite encapsular los diferentes entornos de trabajo, de de esta forma se permite compartir la infraestructura sin que los integrantes de un proyecto puedan acceder a la información de otro.

  Uno de los grandes argumentos detractores para la adopción de esta tecnología por parte de las empresas, es ceder la gestión de lo datos a proveedores externos a la empresa. El hecho de implementar un cloud interno corta de raíz esta problemática y nos permite decidir que procesos de negocio son externalizados y cuales permanecerán únicamente controlados por la empresa.

Disponemos de un entorno de pruebas formado por **2 servidores** capaces de centralizar la gestión de la infraestructura y **4 nodos de trabajo** donde se desplegaran la máquinas virtuales. Se utilizaran para las pruebas con cada uno de los gestores con el objetivo de ampliar el estado del arte iniciado en la sección 5, el objetivo es estudiar la capacidad de cada gestor de afrontar los retos planteados por un entorno de desarrollo y describir sus diferentes funcionalidades a la hora de aprovisionar máquinas virtuales y distribuirlas entre los nodos de trabajo. Se hará especial hincapié en la capacidad de interacción con Amazon EC2, cloud público que también se pretende estudiar, con la finalidad de adquirir experiencia en la utilización de este tipo de entornos.

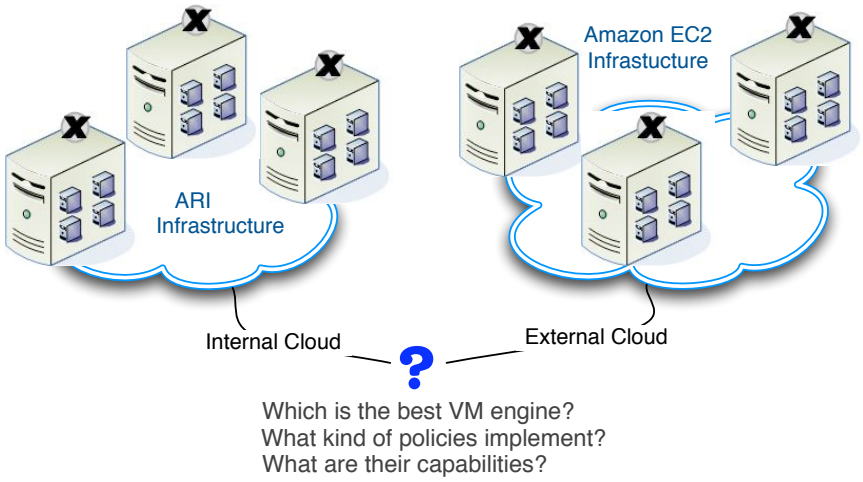


Figura 10: Diseño inicial

8.1. Arquitectura

El **cloud interno/privado** es diseñado con la finalidad de importar los beneficios asociados a las estructuras cloud externas y mitigar los riesgos derivados de ceder a terceros el control tanto de la infraestructura como de los datos.
Para realizar esta implantación de forma incremental, se partirá de una arquitectura genérica de donde se extraerán las funcionalidades y herramientas open-source adecuadas para un entorno de desarrollo.

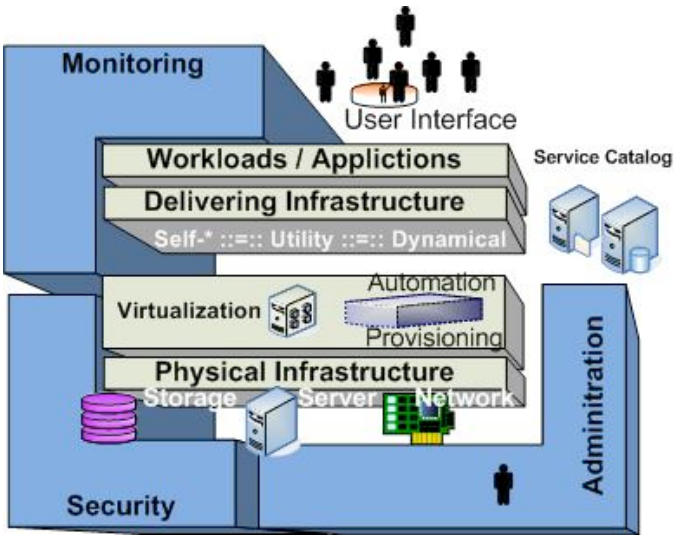


Figura 11: Arquitectura genérica para un cloud interno

La integración de estos sistemas y su arquitectura, dependerá mucho del propósito final de la infraestructura. Con la intención de acotar el entorno de pruebas, se definirá la arquitectura de acuerdo a las funcionalidades de un **entorno de desarrollo**, donde se ofrecerán máquinas virtuales preconfiguradas con las que poder realizar **tests** de integración software.

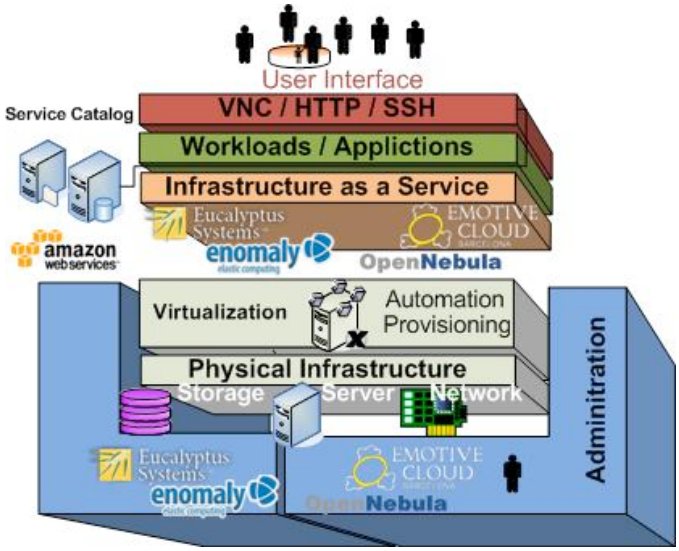


Figura 12: Arquitectura entorno desarrollo

La relación entra cada uno de los componentes de la arquitectura marcará las características final del servicio. La administración del servicio y de los recursos estará totalmente controlada por el departamento de sistemas de IT, los desarrolladores tendrán accesos a su entorno programático y posibilidad de administrar el software dentro de este entorno.

8.2. Implantación

No hay una metodología clara a la hora plantear la adopción de este modelo, no obstante existen una seria de buenas prácticas que adoptar para facilitar la progresiva implantación. rPath[21] describe la adopción en 5 fases (figura 13) para que una organización puede llegar a ofrecer un servicio en el Cloud. Los pasos seguidos durante la implantación de nuestro entorno concuerdan con el modelo de rPath en las dos etapas iniciales. Las posteriores etapas hasta la encapsulación y prestación final del servicio vienen marcadas por la finalidad de nuestro entorno que es ofrecer VM donde poder desarrollar software.

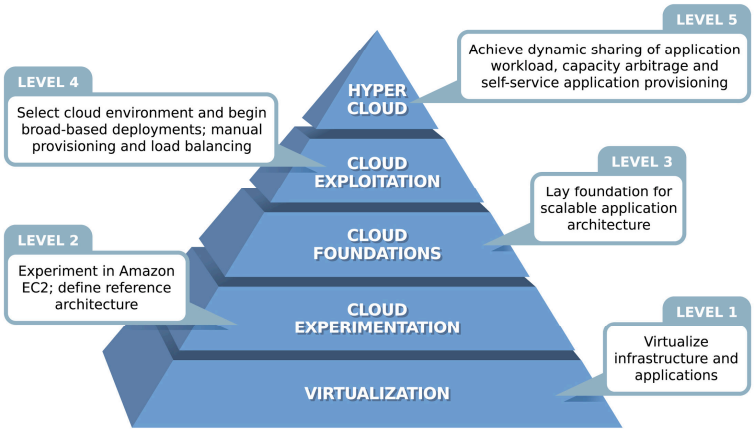


Figura 13: Modelo de adopción del Cloud Computing

A continuación se enuncian las fases seguidas en nuestro proyecto:

- Fase 1: Experimentación con proveedores externos (Amazon EC2).
- Fase 2: Virtualización de los recursos.
- Fase 3: Gestión de la infraestructura desde un punto centralizado.
- Fase 4: Definir políticas del servicio.
- Fase 5: Automatizar y encapsular el servicio.

El proyecto cubre las tres primeras fases, desarrolladas en las secciones posteriores nos ayudarán a experimentar con cada uno de los gestores. Los niveles descritos en la figura 13 se realizan tanto para la infraestructura como para las aplicaciones. En nuestro caso concreto no centraremos únicamente en la prestación de la infraestructura en forma de VM como un servicio.

8.2.1. Fase 1: Experimentación proveedor externo (Amazon EC2)

Es importante dar los primeros pasos en una buena dirección y sin duda este es el camino de amasar conocimientos, obtener comprensión y acumular experiencia en este tipo de tecnología.

Por lo tanto, el siguiente paso lógico implica la experimentación dentro de un public cloud con el fin de sentar las bases para nuestro futuro cloud interno.

La empresa Amazon y sus servicios

Amazon Web Services (AWS) es una creciente gamma de servicios de computación ofrecidos remotamente y sobre Internet por **Amazon.com, Inc**.

Esta multinacional Americana del comercio electrónico lanzo en Julio del 2002 estos servicios sobre Internet. Ofreciendo beneficios tanto a usuarios finales como a desarrolladores, permitiendo reutilizar las funcionalidades que se ofrecían. La misma empresa contaba con el alta en su sistema de 330,000 desarrolladores en 2007 [22], actualmente cuenta con infraestructuras en diferentes zonas de Estados Unidos y Europa.

AWS utiliza las interfaces *Representational state transfer (REST)* y *Simple Object Access Protocol (SOAP)* para sus servicios web (WS), ambos protocolos son utilizados para hacer accesible a Amazon sobre HTTP. Describimos a continuación algunos de estos servicios:

- **Amazon Elastic Compute Cloud (EC2)**, proporciona un entorno escalable de serviodores virtualizados basados en Xen.
- **Amazon Simple Storage Service (S3)**, proporciona un WS basado en el almacenamiento de datos.
- **Amazon SimpleDB**, proporciona las principales funcionalidades de indexación y consulta de una base de datos, permitiendo ejecutar consultas sobre datos estructurados.
- **Amazon Simple Queue Service (SQS)**, proporciona un servicio de gestión de colas, almacenamiento y intercambio de mensajes entre ordenadores en la red.
- **Amazon CloudFront**, proporciona un sistema de de red entre ordenadores (content delivery network - CDN), que permite distribuir los objetos almacenados en S3 cerca de donde se producen las peticiones.

Hay otros servicios proporcionando funcionalidades especificas que dan soporte a la integración con la plataforma comercial amazon.com, *Amazon CloudWatch* proporciona monitorización para los AWS, *Auto Scaling* permite escalar automáticamente la capacidad de acuerdo a unas condiciones establecidas.

Nuestro entorno de desarrollo se beneficiara de los servicios EC2 para proveernos de las imágenes de los sistemas operativos y S3 para almacenar estas imágenes extendiendo la capacidad de nuestra infraestructura interna.

Configuración AWS

Inicialmente, para empezar a trabajar debemos crear una cuenta en AWS y darnos de alta en cada uno de los servicios web con los que interacturemos, en nuestro caso EC2, S3 y AWS Management Console. Una vez las cuentas han sido confirmadas podemos empezar a disponer del servicio, en este punto es necesario proveer un mecanismo de pago, asociando a nuestra cuenta una

tarjeta bancaria. No hay cargos derivados de estas operaciones, los servicios no tienen coste hasta no ser usados.

Con el fin de gestionar de forma segura nuestros servicios se provee un certificado X.509 así como claves de acceso, estos pueden ser gestionados como el servicio de facturación desde el panel de control de que dispone nuestra cuenta.

Gestión

Existen diferentes herramientas para una correcta gestión de nuestro entorno, en esta sección describimos algunas posibles soluciones para cada uno de los servicios.

AWS Management Console

El propio Amazon ofrece *AWS Management Console* una interfaz web con la que interacturar con el servicio EC2, se prevé el soporte para S3 desde la consola de gestión en futuros lanzamientos. Este servicio permite a los desarrolladores arrancar y detener instancias de EC2, realizar acciones sobre estas instancias y controlar configuraciones como grupos de seguridad, almacenamiento de volumenes, claves de acceso a las instancias y configuraciones de red mediante *Elastic IP*.

Línea de comandos

Estas herramientas estan disponible en el Centro de Recursos de Amazon.com, se proporcionan *scripts* tanto para Windows como para Linux capaces de ayudar a los desarrolladores en la gestión de sus propios recursos, existen dos paquetes con diferente cometido.

- Amazon EC2 API:
  Esta herramienta sirve como cliente sobre el servicio web EC2, ofreciendo diversas funcionalidades que nos permiten gestionar las instancias.
- Amazon EC2 AMI:
  Esta utilidad ayuda a cagar y descargar las imágenes con las que trabajamos en S3, entre otras funciones de almacenamiento.

Plugins de firefox

- *Elasticfox* es una extensión del navegador Firefox que nos permite gestionar las imágenes de Amazon EC2 y S3. Este complemento nos permite interactuar gráficamente con nuestro entorno desplegado sobre Amazon. AWS mantiene y distribuye este plugin.

- *S3 Organizer (S3Fox)* es un plugin no oficial, que nos permite organizar, gestionar y almacenar nuestros ficheros en Amazon S3.

Cloud interno interactuando con Amazon EC2

Eucalyptus, La interfaz de Eucalyptus es compatible con Amazon EC2, la intención de este gestor es soportar la interacción con múltiples proveedores en el cloud. EC2 por el momento parece el proveedor mejor documentado entre las opciones existentes, aparte de proporcionar una interacción directa con Eucalyptus ya que este último, comparte tanto en el caso del servicio EC2 como el S3 la misma API que Amazon ofrece para gestionar su infraestructura.

Enomaly, ECP asegura soportar múltiples proveedores en el cloud, incluyendo Amazon Elastic Compute Cloud. No obstante este proyecto no realizará esta interacción debido a que la excasa documentación ofrecida para la versión abierta del producto en la comunidad de desarrollo.

Globus Nimbus, *Nimbus EC2 fontend* es una implementación de la descripción del servicio web de Amazon EC2 (WSDL) que nos permite utilizar clientes desarrollados por el sistema de Amazon contra los clouds basados en Nimbus.

OpenNebula, Proporciona *plugins* de acceso a Amazon EC2 para complementar los recursos locales que gestiona, trata a Amazon como a una infraestructura más de la que proveerse. Este gestor no proporciona interacción alguna con el sistema de almacenamiento de instancias de Amazon, ni monitorización sobre la instancias.

Configuración del entorno

Está sección cubrirá los pasos necesarios para trabajar con los servicios EC2 y S3 con nuestras propias instancias modificadas. Antes de empezar es necesario introducir algunos de los conceptos con los que trabajaremos.

- **Amazon Machine Image (AMI),** Es como Amazon denomina a sus imágenes Xen, se trata de unos ficheros encriptados almacenados en S3 a la espera de ser desplegados. Estos ficheros contienen toda la información necesaria para arrancar las máquinas virtuales.
- **Instancia,** Los sistemas en ejecución basados en una AMI son denominados instancias.

Definiremos el proceso a seguir por el administrador del entorno para que nuestro desarrollador pueda empezar a programar.

1	Configurar el entorno de trabajo: JRE, comandos de interacción con EC2 y S3, variables de entorno, claves de acceso.
2	Crear una AMI con todo el software: el sistema operativo, las configuraciones de este, aplicaciones, librerias, etc.
3	Cargar esta AMI a S3
4	Registrar la AMI con EC2: Esto permite verificar que el paso 3 se ha realizado correctamente se nos retorna un identificador para la AMI.
5	Utilizar el identificador retornado y las APIs que proporciona el servicio web, para ejecutar, monitorizar y finalizar tantas instancias de la AMI como sean necesario.

Cuadro 4: Proceso de creación y utilización AMI.

El proceso de creación de nuestra AMI lo podemos realizar a partir de alguna de las imágenes base que proporciona el servicio. Una vez desplegada y instalado todo el software procedemos a empaquetar la que será nuestra nueva imagen base.

```
ec2-bundle-vol -d <mount_directory> -k <private_key> \
-c <cert-key> -u <user_id> -p <manifest_file>
```

Esta acción nos creará una archivo XML que contiene la relación entre los ficheros o partes de nuestra nueva AMI. Utilizando este archivo XML podemos almacenar nuestra imagen en S3 para poder desplegarla sobre EC2.

```
ec2-upload-bundle -b <bucket_name> -m <manifest_file.xml>  \
-a <access_key> -s <secret_key>
```

Una vez almacenada, antes de poder instanciar nuestra imagen debemos registrarla. Esto nos retornará un identificador para la imagen y nos confirmará que ha sido almacenada correctamente.

```
ec2-register <manifest_file.xml>
```

Con el identificador podemos realizar operaciones sobre nuestra imagen registrada, a continuación se muestran los pasos que nos permitirán desde instanciar nuestra imagen hasta terminar su ejecución.

```
# Muestra todas las imagenes publicas
ec2-describe-images --all
# Muestra mis imagenes
ec2-describe-images
    IMAGE <ami-id> ...
```

```
# Muestra caracteristicas de la instancia
ec2-describe-instances
# Muestra zonas donde desplegar instancia
ec2-describe-availability-zones

# Empezamos a instanciar
ec2-add-keypair key > key.private
chmod 0600 mykey.private
ec2-run-instances -k key -n <number of instances to start> \
<ami-id>
# Autorizar conexiones ssh
ec2-authorize -P tcp -p 22 -s 0.0.0.0/0 default
# Nos conectamos a la instancia
ssh -i key.private root@<accessible-instance-ip>
# Terminar instancia
ec2-terminate-instances <instance-id1> ... <instance-idn>
```

8.2.2. Fase 2: Virtualización de los recursos

Xen es un hipervisor o VMM paravirtualizado, originalmente desarrollado como parte del proyecto XenoServers por la Universidad de Cambridge. Lanzo la primera versión (1.0) en 2003 y en la actualidad su madurez lo hace capaz de afrontar diferentes escenarios en entornos de producción [23].

Gracias a la paravirtulización, Xen es capaz de ejecutar múltiples máquinas virtuales en un mismo sistema físico sin un soporte específico del hardware, consiguiendo un alto rendimiento en las arquitecturas X86 (32/64 bits) tradicionalmente difíciles de virtualizar.

Xen es un proyecto abierto, la mayor parte de Xen rige bajo la licencia GNU GPL, vervsion 2, otras partes estan licenciadas bajo GLPL, ZPL 2.0 o licencias *BSD-style*. El VMM paravirtualizado esta desarrollado para trabajar sobre GNU/Linux todo y soportar otros SO como NetBSD, FreeBSD y Solaris. Una de las misiones del proyecto, es instaurarse como hipervisor opensource estándar en la industria, siendo capaz de ofrecer sus funcionalidades para diferentes familias de procesadores(e.g. Pentium Pro, Celeron, Pentium II, Pentium III, Pentium IV, Xeon, AMD Athlon, AMD Duron).

Arquitectura

Las máquinas virtuales ejecutadas por Xen reciben el nombre de **dominios**. La topología de estos sistemas esta formada por capas, la capa inferior es la que goza de mayores privilegios y es donde se instala el hipervisor de Xen ejerciendo de anfitrión [24].

Las capas trabajan conjuntamente para ofrecer un entorno de virtualización consistente. La organización general de estas capas se muestra en la figura 14.

A continuación se realiza un breve repaso de cada una:

- **Hipervisor Xen,** Capa de abstracción software que se encuentra sobre el hardware del sistema. Es el encargado de la planificación de la CPU y particionado de memoria para las máquinas virtuales que residen por encima del hipervisor.
- **Dominio 0 (dom0),** Este dominio es una modificación del núcleo de linux ejecutado sobre el hipervisor, dispone de privilegios especiales que le permiten acceder a los dispositivos de E/S así como interactuar con otras máquinas virtuales en el sistema. En el dom0 se incluyen dos drivers (Network Backend Driver, Block Backend Driver) capaces de procesar las peticiones por parte de los domU a la red y los discos.
- **Dominio de Gestión y Control (Xen DM&C),** Colección de demonios utlizados para la gestión y el control del entorno virtualizado desde el dom0.
- **Dominios paravirtualizados (DomU),** Máquinas paravirtuales ejecutadas sobre Xen. Por si solas no tienen accesos a los recursos hardware o a otros dominios ejecutados en el mismo sistema. Este dominio huésped contiene dos drivers para acceder al los discos y la red (Block Driver, Network Driver) .

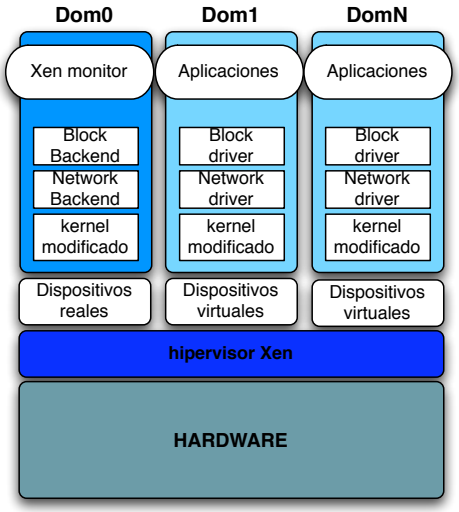


Figura 14: Arquitectura Xen paravirtualizado

Xend (node control software)

El demonio de Xen (xend) realiza las funciones de gestión del sistema, actuando como punto de control central de la máquina. Debe estar en ejecución para poder arrancar y gestionar las máquinas virtuales. Este demonio

es ejecutado por el root del sistema ya que necesita acceso privilegiado al sistema.

Los eventos son capturados y almacenados en /var/log/xend.log, la configuración de este demonio esta supeditada al archivo /etc/xen/xend-config.sxp desde el que se describe esencialmente la configuración de red entre el sistema físico y los dominios virtuales.

Nos encontramos en una red local con direcciones privadas y disponemos de varias direcciones IP a la espera de ser asignadas a nuestras VM. Para que Xen trabaje en modo puente (bridge) debemos tener descomentadas las siguientes lineas en el archivo de configuración:

```
# -*- sh -*-

#
# Xend configuration file.
#
(xend-http-server yes)
(xend-unix-server yes)
# To bridge network traffic, like this:
#
# dom0: fake eth0 -> vif0.0 -+
#                            |
#                          bridge -> real eth0 -> the network
#                            |
# domU: fake eth0 -> vifN.0 -+
(network-script network-bridge)
(vif-script vif-bridge)
# dom0-min-mem is the lowest memory level
# (in MB) dom0 will get down to.
(dom0-min-mem 256)
# If dom0-cpus = 0, dom0 will take all cpus available
(dom0-cpus 0)
```

Configuración del dominio

La creación de los dominios Xen se puede realizar mediante el paquete **xen-tools**, donde se incluyen diversos scripts para facilitar la creación de nuevos dominios. La máquina virtual una vez creada consta de una imagen de un sistema de ficheros (.img), un imagen del swap (opcional) y el archivo donde consta la configuración de este dominio.

Este archivo de configuración permite describir con más detalle las especificaciones de red, los controladores del dominio y el comportamiento del dominio entre otras fucionalidades. Veamos algunas de las especificaciones básicas:

```
#
# Nucleo & Memoria
```

```
#
kernel = '/boot/vmlinuz-2.6.24-17-xen'
ramdisk = '/boot/initrd.img-2.6.24-17-xen'
#kernel = 'vmlinuz-2.6.24-16-generic'

#
# Virtualizacion por hardware
#
#builder = "hvm"

#
# Memoria
#
maxmem = '1024'
memory = '1024'

#
# Nombre host
#
name = "testVM"
#128-bit UUID
uuid = "709cb535-41a1-a361-7865-6538555af02b"


#
# Procesador
#
#Numero de procesadores
vcpus=3
#cpus = "0,1,2"

#
# Red
#
#vif = [ 'ip=192.168.252.47,mac=00:15:3E:EF:CA:22' ]
# DHCP
#vif = ['bridge=xenbr0']     #if in only dhcp mode
vif = ['ip = "192.168.252.47", bridge=xenbr0']

#
# Discos
#

root = '/dev/xvda ro'
disk = [ "tap:aio:/path/to/image/test-VM-base.img,hda,w" ]

#
# Comportamiento
#
```

```
bootloader = "/usr/bin/pygrub"

on_poweroff = 'destroy'
on_reboot = 'restart'
on_crash = 'restart'
```

Modificación del dominio

Una vez creada la imagen del nuevo dominio (y el fichero swap y de configuración del dominio), Xen permite la gestión del entorno a través de una consola de comandos utilizando la instrucción xm (Xen management user interface).

```
# Iniciar VM
xm create <config_file.cfg>
# Listar dominios
xm list
# Abrir consola al dominio
xm console <domain_name>
```

Una vez iniciada la máquina virtual podemos realizar modificaciones sobre la misma, no obstante no es estrictamente necesario arrancar un dominio para modificar el contenido de nuestra VM. El fichero *domain-name.img* consiste en la imagen de un sistema de ficheros y somos capaces de modificar su contenido, esto es muy útil por ejemplo cuando intentamos trabajar con imágenes inconsistentes.

```
# Montamos el sistema de ficheros
mount -o loop /path/to/image/test-VM-base.img /mnt
# Cambiamos el directorio de root
# Para interactuar con la imagen
chroot /mnt
```

8.2.3. Fase 3: Gestión de la infraestructura

En esta fase ya se ha adquirido los conocimientos base sobre este tipo de entornos, gracias a la experimientación con EC2 y con la técnología de virtualización Xen. Con nuestro entorno ya virtualizado la fase 3 analiza diferentes gestores capaces de manejar nuestro entorno de pruebas.

Open Nebula

OpenNebula ha sido desarrollado por el *Distributed Systems Architecture Research Group* de la Universidad Complutense de Madrid y parcialmente financiado por el proyecto europeo *RESERVOIR -Resources and Services Virtualization without Barriers.*

OpenNebula bajo licencia GPL-version 2 ejerce como gestor abierto de un entorno virtualizado, permitiendo el aprovisionamiento y recolocación de máquinas virtuales entre los nodos que forman la infraestructura. En su versión más reciente 1.4, ofrece soporte para tres de los hipervisores más adoptados Xen, KVM, VMWare. Nuestras pruebas para el entorno de desarrollo han sido realizadas con la anterior versión estable 1.2, donde se soporta tanto Xen como KVM.

Arquitectura

La arquitectura interna mostrada en la figura 15 muestra los tres pilares del gestor OpenNebula [25].

- Herramientas
- Core del gestor
- Drivers

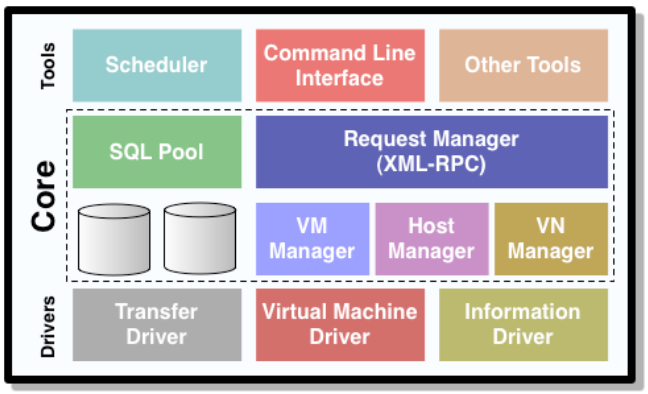


Figura 15: Arquitectura Open Nebula

Herramientas

La primera capa contiene las herramientas necesarias para la gestión de la infraestructura. Una **linea de comandos (CLI)** que permite manipular la infraestructura virtual.

- **onevm:** despliegue, control y monitorización de máquinas virtuales.
- **onehost:** añadir, eliminar y monitorizar nodos.
- **onevnet:** añadir, eliminar y monitorizar redes virtuales.

El *scheduler* o planificador ha sido diseñado como una entidad independiente de la arquitectura, esto permite que pueda ser fácilmente desacoplado del resto de los componentes. Utiliza una interfaz XML-RPC para invocar acciones sobre las máquinas virtuales. Su función es asignar las VM a los nodos que cumplan los requerimientos de capacidad solicitados por la VM. Los recursos que no cumplen los objetivos son filtrados, en este caso un proceso de ranking es ejecutado para seleccionar el mejor recurso disponible. El ranking puede ser definido en el *template* de creación de la VM.

El scheduler esta construido sobre un *template* que define sus políticas, este *template* puede ser usado para adaptar el comportamiento del planificador. El planificador implementa el algoritmo *Match-making* (mm_sched) que es el encargado de priorizar los recursos más convenientes para cada VM. En primer lugar comprueba que nodos cumplen las condiciones fijadas en el atributo *REQUERIMENTS* en el archivo de creación. El atributo *RANK* es el encargado de definir que política sigue el planificador de entre las siguientes:

- **Packing Policy:** Utiliza los nodos en que hay más VM ejectuandose.
- **Striping Policy:** Utiliza los nodos en que hay menos VM ejecutandose.
- **Load-aware Policy:** Utiliza los nodos con más CPU libre.

Haizea lease manager (http://haizea.cs.uchicago.edu) , en combinación con Open Nebula ofrece funcionalidades extra de planificación. Permite realizar reservas avanzadas de los recursos virtuales.

- **Advance reservation:** Permite definir la hora exacta de arrancada de un VM.
- **Best-effort provisioning:** Aprovisiona la VM tan pronto como es posible dependiendo de los recursos disponibles, la petición queda en una cola si no hay recursos disponibles.
- **Immediate provisioning:** Aprovisiona la VM al momento.

Core del gestor

El core de Open Nebula consiste en un juego de componentes que permiten controlar y supervisar las máquinas virtuales y físicas. Este realiza sus acciones invocando al driver correspondiente, la comunicación entre los drivers y el demonio oned es realizada mediante un protocolo de mensajes ASCII (Unix Pipes-diver message protocol). Los principales componentes son:

- **Request Maganer:** Realiza el manejo de las peticiones, se expone mediante una interfaz XML-RPC comunicando las peticiones recibidas a través de línea de comandos o por parte del planifcador al core de OpenNebula, dependiendo del método invocado un componente es llamado internamente.
- **Virtual Machine Maganer:** Es el responsable de la gestión y monitorización de la VM.
- **Virtual Network Manager:** Es el encagado de manejar el uso de las direcciones de red, permite la creación de redes virtuales formadas por un cojunto de dirección IP y MAC.
- **Host Manager:** Realiza la gestión y monitorización de las máquinas físicas.
- **Database:** Basada en SQLITE3 es el componente principal encargado de la estructura de datos interna. Este componente asegura la adaptabilidad y fiabilidad de la estructura de las VMs, en caso de fallo permite recuperar esa estructura rápidamente.

Drivers

Dispone de un conjunto de conexiones a módulos que nos permiten interactuar recíprocamente con un middleware específico. Esta formado por:

- **Transfer Driver:** Realiza el despliegue de la imágenes, clonación, borrado, creación de swap.
- **Virtual Machine Driver:** Actua sobre el estado de las máquinas virtuales, arrancada, parada, migración.
- **Information Driver:** Obtiene la información de las máquinas físcias, memoria, cpu, disco.

Ciclo de vida del aprovisionamiento

- **Pending:** Después de indicar la creación de la VM, la descripción se almacena en la BBDD. El planificador es el encargado de encontrar el nodo más adecuado para esta descripción.
- **Prolog:** Durante este estado la VM se esta tranfiriendo desde el repositorio hasta el nodo donde será desplegada. En este estado se realiza una clonación de la máquina del repositorio y se crear la imagen de disco swap especificado en la descripción.

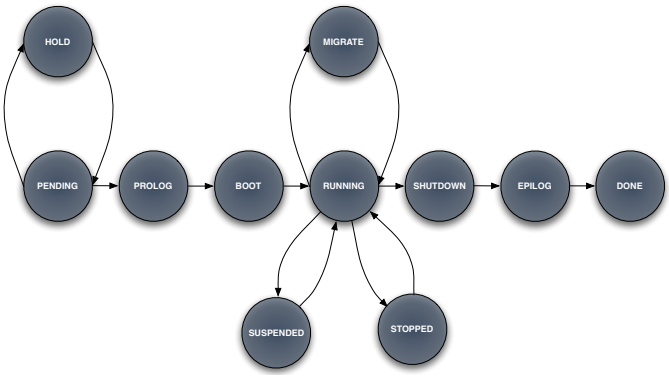


Figura 16: Estados en el aprovisionamiento de la VM.

- **Boot:** Se genera el archivo de configuración para el hipervisor donde la VM ha sido desplegada. Y se inicia el dominio.
- **Running:** En este estado la VM esta en fucninamiento, es monitorizada periódicamente para conocer la cantidad de recursos que consume.
- **Shutdown:** Se apaga la ejecución de la VM.
- **Epilog:** Copia en el repositorio las VM con el atributo SAVE=yes. Elimina la imagen del disco y swap.

```
[root@erriapo templates]# head -40 /var/log/one/53.log
Wed Jul 29 14:46:38 2009 [DiM][I]: New VM state is ACTIVE.
Wed Jul 29 14:46:39 2009 [LCM][I]: New VM state is PROLOG.
Wed Jul 29 14:46:40 2009 [TM][I]: tm_clone.sh: erriapo:/srv/nebula/images/vm-test/test.img containerdraft:/srv/nebula/images/53/images/disk.0
Wed Jul 29 14:46:40 2009 [TM][I]: tm_clone.sh: DST: /srv/nebula/images/53/images/disk.0
Wed Jul 29 14:46:40 2009 [TM][I]: tm_clone.sh: Creating directory /srv/nebula/images/53/images
Wed Jul 29 14:46:40 2009 [TM][I]: tm_clone.sh: Executed "ssh containerdraft mkdir -p /srv/nebula/images/53/images".
Wed Jul 29 14:46:40 2009 [TM][I]: tm_clone.sh: Cloning erriapo:/srv/nebula/images/vm-test/test.img
Wed Jul 29 14:53:08 2009 [TM][I]: tm_clone.sh: Executed "scp erriapo:/srv/nebula/images/vm-test/test.img containerdraft:/srv/nebula/images/53/images/disk.0".
Wed Jul 29 14:53:09 2009 [TM][I]: tm_clone.sh: Executed "ssh containerdraft chmod a+w /srv/nebula/images/53/images/disk.0".
Wed Jul 29 14:53:09 2009 [TM][I]: tm_mkswap.sh: Creating 1024Mb image in /srv/nebula/images/53/images/disk.1
Wed Jul 29 14:53:10 2009 [TM][I]: tm_mkswap.sh: Executed "ssh containerdraft mkdir -p /srv/nebula/images/53/images".
Wed Jul 29 14:53:11 2009 [TM][I]: tm_mkswap.sh: Executed "ssh containerdraft dd if=/dev/zero of=/srv/nebula/images/53/images/disk.1 bs=1 count=1 seek=1024M".
Wed Jul 29 14:53:11 2009 [TM][I]: tm_mkswap.sh: Initializing swap space
Wed Jul 29 14:53:12 2009 [TM][I]: tm_mkswap.sh: Executed "ssh containerdraft /sbin/mkswap /srv/nebula/images/53/images/disk.1".
Wed Jul 29 14:53:13 2009 [TM][I]: tm_mkswap.sh: Executed "ssh containerdraft chmod a+w /srv/nebula/images/53/images/disk.1".
Wed Jul 29 14:53:13 2009 [LCM][I]: New VM state is BOOT
Wed Jul 29 14:53:13 2009 [VMM][I]: Generating deployment file: /var/lib/one/53/deployment.0
Wed Jul 29 14:53:13 2009 [VMM][I]: Command: scp /var/lib/one/53/deployment.0 containerdraft:/srv/nebula/images/53/images/deployment.0
Wed Jul 29 14:53:14 2009 [VMM][I]: Copy success
Wed Jul 29 14:53:14 2009 [VMM][I]: Setting credits for the VM
Wed Jul 29 14:53:14 2009 [VMM][I]: Command: sudo /usr/sbin/xm create /srv/nebula/images/53/images/deployment.0 \&\& sudo /usr/sbin/xm sched-cred -d one-53 -w
128
Wed Jul 29 14:53:16 2009 [LCM][I]: New VM state is RUNNING
Wed Jul 29 14:53:47 2009 [VMM][I]: Monitor Information:
       CPU   : 0
       Memory: 130824
       Net_TX: 0
       Net_RX: 166
```

Figura 17: Aprovisonamiento de una VM en Open Nebula.

Configuración entorno

Nuestro entorno de pruebas para este gestor esta formado por una máquina (Master), donde estará instalado ONE el motor de gestión, y de 2 a 4 nodos donde se desplegarán las máquinas virtuales Xen.

El sistema operativo elegido en nuestro entorno tanto para el master como para los nodos es Centos5.2. OpenNebula esta desplegado sobre el directorio raíz (system wide). La gestión de usuarios se realiza mediante un servidor NIS que permite tratar a los usuarios de una forma centralizada desde el master. Estos usuarios establecen un sistema de confianza entre los diferentes recursos por medio de claves SSH. El mismo master actúa de centro de monitorización y de repositorio de imágenes, existe la posibilidad de realizar el almacenamiento de imágenes en un repositorio externo y gestionarlo mediante un servidor NFS, no obstante nuestras pruebas se basarán en el aprovisionamiento con el fin de consolidar la capa de virtualización. Se ofrece la posibilidad de aprovisionar VM tanto a los nodos como desplegar máquinas sobre EC2.

Para que el master trabaje de forma adecuada con el hipervisor Xen se debe indicar en el archivo de configuración del demonio oned (etc/one/oned.conf).

```
####
#Configuracion de los drivers
####
# Information Driver
IM_MAD = [
    name       = "im_xen",
    executable = "/usr/lib/one/mads/one_im_ssh",
    arguments  = "im_xen/im_xen.conf",
    default    = "im_xen/im_xen.conf" ]

# VM Driver
VM_MAD = [
    name       = "vmm_xen",
    executable = "/usr/lib/one/mads/one_vmm_xen",
    default    = "vmm_xen/vmm_xen.conf",
    type       = "xen" ]

# Transfer Driver
TM_MAD = [
    name       = "tm_ssh",
    executable = "/usr/lib/one/mads/one_tm",
    arguments  = "tm_ssh/tm_ssh.conf",
    default    = "tm_ssh/tm_ssh.conf" ]
```

Configuración OpenNebula

El master necesita saber como acceder y como utilizar cada uno de los nodos, a la hora de crearlos se debe indicar el nombre de máquina y los drivers asociados.

```
onehost create nodeHOST im_xen vmm_xen tm_ssh
```

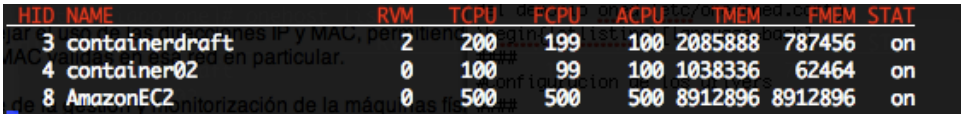


Figura 18: Monitorización nodos disponibles.

En nuestra red estática las parejas de direcciones IP y MAC deben ser definidas explícitamente. Esto se muestra en un template de creación de la red virtual donde se indica que interfaz actuará a modo de puente para garantizar el acceso a internet.

```
NAME   = "Test_VLAN"
TYPE   = FIXED
BRIDGE = xenbr0
LEASES = [IP=192.168.X.X,MAC=XX:XX:XX:XX:XX:XX]
```

Creamos la red virtual mediante el comando onevnet.

```
# Creacion de la red
onevnnet create Test-template.net
# Monitorizacion de la red
onevnet show "Test_VLAN"
    NID              : 8
    UID              : 0
    Network Name     : Test VLAN
    Type             : Fixed
    Bridge           : xenbr0

     ....:  Template  :....
        BRIDGE=xenbr0
         LEASES=IP=192.168.X.X,MAC=XX:XX:XX:XX:XX:XX
            NAME=Test VLAN
            TYPE=FIXED

     ....:  Leases  :....
    IP = 192.168.X.X MAC = XX:XX:XX:XX:XX:XX USED = 1 \
    VID = 53
```

La creación de la VM al igual que en el caso de Xen viene definida por un archivo de configuración donde se indican las características de la máquina.
- Configuración:

```
NAME   = vm-test
CPU    = 0.5
MEMORY = 128
OS     = [
  kernel    = "/boot/vmlinuz-2.6.18-6-xen-686",
  initrd    = "/boot/initrd.img-2.6.18-6-xen-686",
  root      = "sda1" ]
DISK   = [
```

```
    source   = "/srv/nebula/images/vm-test/test.img",
    target   = "sda1",
    readonly = "no" ]
  DISK   = [
    type     = "swap",
    size     = 1024,
    target   = "sdb"]
  NIC    = [ NETWORK = "Test VLAN" ]
```

- Creación:

```
onevm create testVM.template
```

Aprovisionamiento sobre Amazon EC2

El driver de OpenNebula para EC2 proporciona la posibilidad de lanzar instancias sobre Amazon, no se tienen acceso sobre el dom0 de EC2, se necesitan herramientas externas para monitorizar el estado de la AMI, otras funcinalidades como almacenamiento o migración tampoco son manejadas por medio del driver.

- Captura de pantalla del entorno de trabajo:

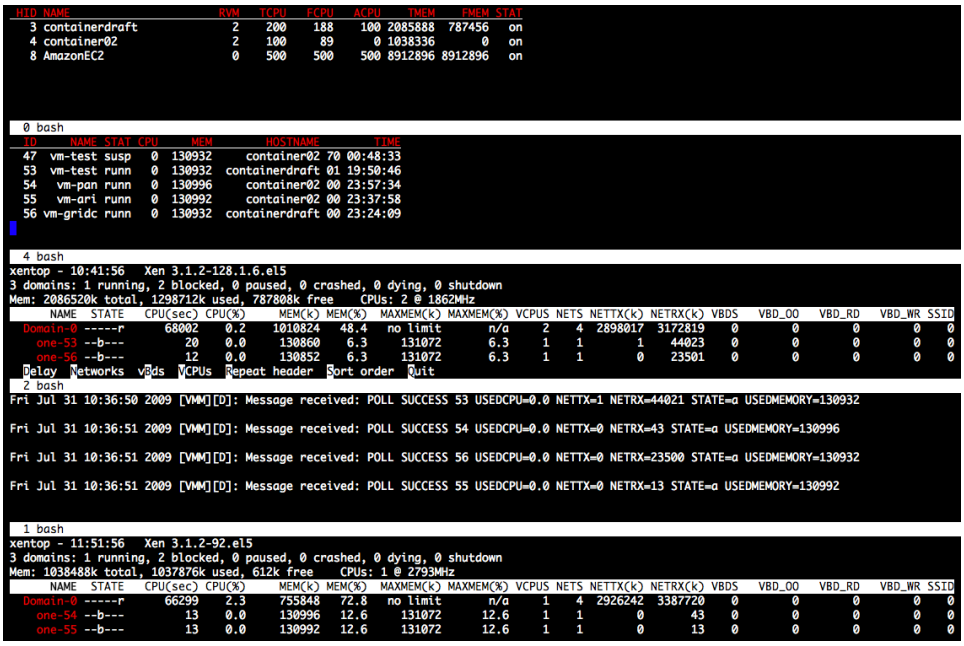


Figura 19: Entorno de trabajo Open Nebula

Eucalyptus

Ha sido desarrollado en el laboratorio MAYHEM por el Departamento de Ciencia y Computación de la Universidad californiana de Santa Barbara (UCSB). Inicialmente nació como entorno para la investigación de la propia universidad, recientemente han iniciado un proceso de comercialización del proyecto, fundando Eucalyptus Inc. Eucalyptus es distribuido abiertamente bajo licencia FreeBSD. Este gestor permite el aprovisionamiento de imágenes virtuales Xen, implementando la misma funcionalidad que la interfaz de Amazon [26].

Nuestro entorno de pruebas se realizara con la versión 1.5.1, última versión estable al inicio de las configuraciones. Esta versión soporta la paravirtualización con Xen. Actualmente se encuentra en la versión 1.5.2 donde se han realizado mejoras en el sistema de almacenamiento, soporte para la virtualización asistida por hardware con KVM e incluye una nueva herramienta de administración (euca_tools) que sustituye a las de EC2.

Arquitectura

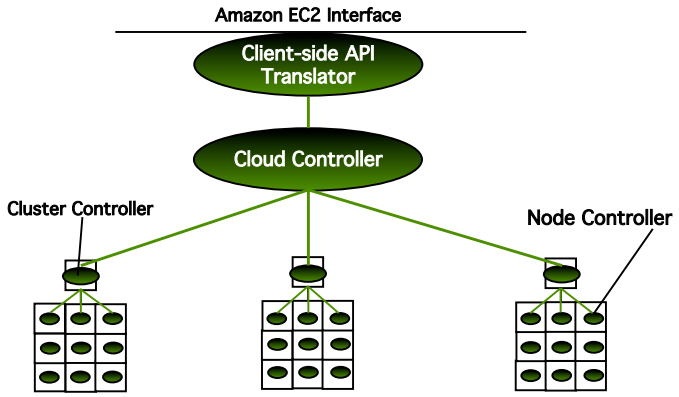


Figura 20: Arquitectura Eucalyptus.

Cloud Controller

El Cloud Controller (CLC) proporciona la interfaz desde donde el administrador y los usuarios pueden interactuar con la infraestructura. El CLC es una colección de servicio web capaces de gestionar los recursos virtualizados subyacentes. Están agrupados en tres categorias:

- **Resource Services,** permite a los usuarios modificar las propiedades de las VM y la red, y monitorizar tanto los recursos físicos del sistema como los recursos virtuales.

- **Data Services,** maneja los datos del sistema y de los usuarios de forma persistente. Proporciona un entorno de usuario configurable según las peticiones de recolocación.
- **Interface Services,** presenta una interfaz exponiendo las herramientas proporcionadas por el gestor.

Este componente se comunica con el Cluster Controller y realiza las elecciones de colocacción de las nuevas instancias. A este nivel se accede a toda la información referente a los usuarios que están ejecutando las instancias, a la VM que se esta ejecutando, así como la monitorización de la carga del sistema.

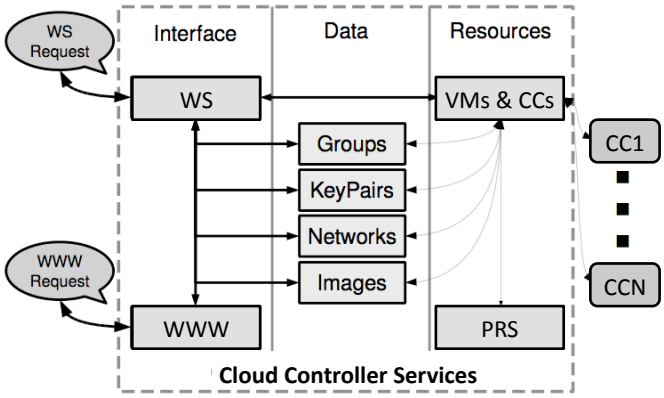


Figura 21: Esquema Cloud Controller

Cluster Controller

El Cluster Controller (CC) trabaja entre el cloud (CLC) y los diferentes nodos (NC). Es el encargado de recibir las peticiones de aprovisionamiento de las VM provenientes del CLC y decidir donde desplegarlas en base al estado de los nodos. La comunicación es bidireccional y el CC en caso de sobre-capacidad puede informar al CLC para que decida aprovisionar las VM en otro CC.

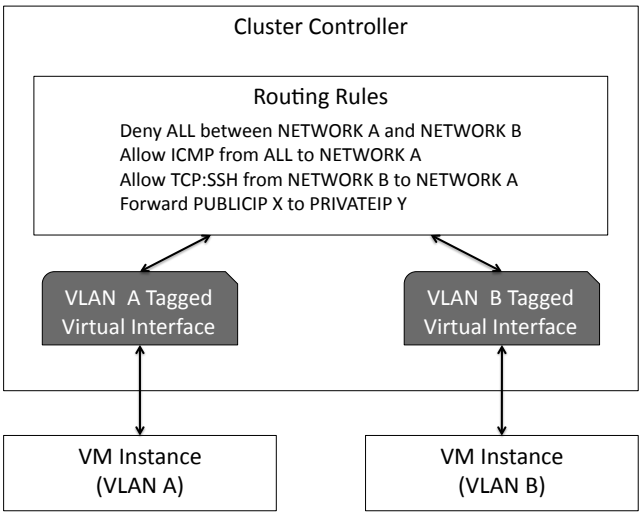


Figura 22: Esquema Cluster Controller.

Node Controller

El Node Controler (NC) se ejecuta sobre las máquinas físicas donde desplegamos las VM. Su función es interactuar con el SO y el VMM. Es el encargado de obtener la información de los recursos físicos presentes en la máquina, como espacio en disco, número de cores y memoria. La información obtenida puede ser consultada por el CC que tomará la decisión final de aprovisionar la VM sobre esos recursos físicos. El NC cuando recibe la petición realiza el despliegue de la máquina virtual y cualquier acción futura sobre ella.

Walrus Storage Controller

El Walrus Storage Controller (WS3) implementa las APIs REST (Representational State Transfer) y SOAP (Simple Object Access Protocol) las cuáles son compatibles con el sistema de almacenamiento de Amazon, Simple Storage Protocol (S3). Proporciona una servicio de almacenamiento (Axis2, Mule) mediante put/get capaz de acceder y almacenar la imágenes virtuales y los datos de los usuarios.

Sus funciones son:

- Almacenamiento de la imágenes de la VM que podrán ser instanciadas en nuestra infraestructura.
- Controla el acceso y almacenamiento de los datos.

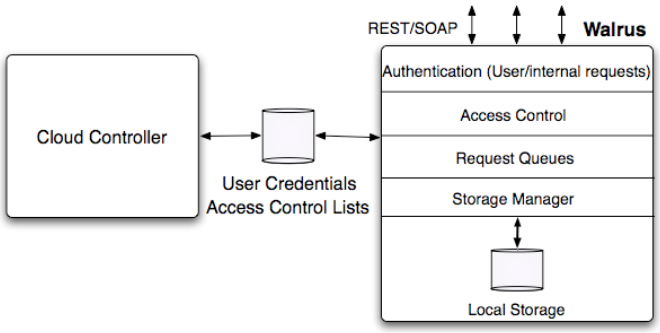


Figura 23: Esquema Walrus Storage Controller

Los usuarios pueden usar este servicio de almacenamiento de la misma manera que el servicio S3 de Amazon, ya que este utiliza la misma herramientas (AMI tools). Este servicio de almacenamiento se ejecuta en la misma máquina que el CLC. El CLC es el encargado de manejar la peticiones hacia Walrus protegido mediante credenciales de usuario y listas de control de acceso.

Elastic Block Storage Controller

El Elastic Block Storage Controller (EBS) se ejecuta en la misma máquina que el Cluster Controllers (CC). Permite crear dispositivos de almacenamiento en bloques persistentes, que pueden ser montados sobre las máquinas en ejecución con el fin de controlar el acceso al disco duro virtual. Se puede crear un sistema de ficheros sobre los EBS o tratarlos de la misma manera que un dispositivo de bloque.

Proporciona la posibilidad de realizar copias de los volúmenes, que son almacenadas en WS3. Estas capturas pueden ser utilizadas para instanciar tantos nuevos volúmenes con se desee. A nivel de red el EBS y las imágenes en los nodos deben estar en el mismo segmento Ethernet. Al dispositivo de bloque se accede utilizando ATA sobre Ethernet (AoE).

Ciclo de vida del aprovisionamiento

- **Verificar la petición:** La autentificación se realiza mediante certificados X509 al igual que EC2. Eucalyptus utiliza claves criptográficas para asegurar la comunicación entre las diferentes partes del sistema, estas comunicaciones están basadas en el estándar WS-Security.
- **Verificar la disponibilidad:** El mismo NC es capaz de obtener la información del recurso físico y verificar la disponibilidad en su sistema

según las características de la VM.

- **Obtención de la imagen:** Tras las verificaciones se obtiene la imagen de la cache local en caso de que la VM ya hubiera sido instanciada o de un repositorio. Eucalyptus trabaja con Walrus como sistema de almacenamiento de donde se puede descargar el núcleo de la máquina, el sistema de ficheros raiz, y opcionalmente la imagen de ram.
- **Crear la red:** Crea la interfaz de red indicada en la configuración asociada a la VM. Esta acción dependerá de el modo de red en que Eucalyptus ese trabajando (STATIC en nuestro caso).
- **Iniciar la VM:** El NC actua sobre le hipervisor para iniciar un nuevo dominio sobre el recuso físico elegido.

Configuración del entorno

Nuestro entorno de prueba para Eucalyptus esta formado por una máquina master, donde se encuentra el Cloud Controller (CLC), el Cluster Controller (CC) y el sistema de almacenamiento Walrus (WS). Nuestro clúster estará formado por 2 nodos con un Node Controller (NC) en cada uno, donde se desplegarán las máquinas virtuales basadas en Xen.

El sistema operativo elegido en este entorno es Centos5.2, tanto en el master como en los nodos. Se ha utilizado la versión 1.5.1 de Eucalyptus, última versión estable al inicio de las configuraciones. Adicionalmente, y debido al interés que Ubuntu a mostrado por Eucalyptus, forjando una alianza que convierte a Eucalyptus en el gestor recomendado por esta distribución, se han realizado pruebas sobre Ubuntu9.04, de la última versión lanzada por Eucalyptus 1.5.2. No obstante, la interacción entre Eucalyptus instalado sobre Ubuntu y los nodos en Centos5.2 con la versión anterior del NC, no se realiza de forma adecuada debido a que el master espera unas rutas de directorios que en los nodos se encuentran en otro lugar, esto se puede solventar encontrando que ruta falla y done esta emplazada, para modificarla mediante un enlace simbólico.

La gestión de usuarios la implementa Eucalytus internamente, gestionada vía web. Inicialmente, el administrador será nuestro único usuario del sistema, hasta que el entorno y sus funcionalidades estén consolidados. La interacción del administrador con nuestra infraestructura interna se realiza mediante las mimas APIs descritas en la experimentación con Amazon (API y AMI tools), de este modo la gestión con nuestro proveedor externo no implica que el administrador aprenda nuevas herramientas para poder utilizarlo.

Para el correcto funcionamiento del gestor existen una serie de dependencias para el master y cada uno de los nodos. En el master se debe instalar Java6 y ruby para hacer funcionar las API que nos permiten gestionar la

infraestructura. Se necesita Apache Ant para ejecutar el CLC. En los NC se invocan scripts en Perl.

Al igual que OpenNebula existe un archivo de configuración (eucalyptus.conf) que definirá la forma en que trabaja Eucalyptus.

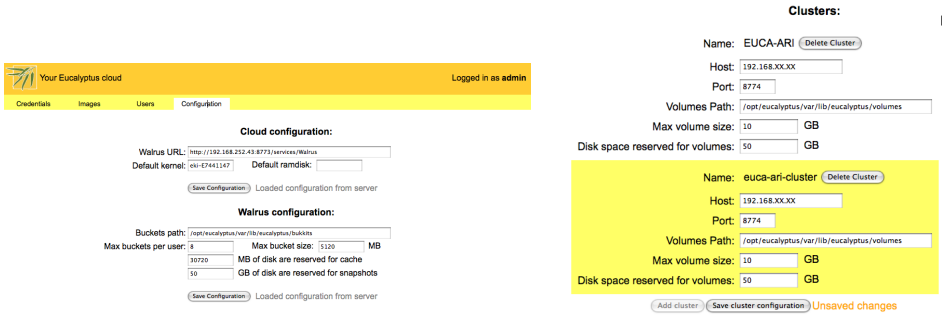
```
# Puertos CLC
CLOUD_PORT="8773"
CLOUD_SSL_PORT="8443"
# Puerto CC
CC_PORT="8774"
# Politica planifciacion (GREDDY,ROUNDROBIN)
SCHEDPOLICY="GREEDY"
# Tecnologia de virtualizacion
HYPERVISOR="xen"
# Lista de nodos
NODES="192.168.XX.XX, 192.168.XX.XX"
# Configuraciones de red
VNET_INTERFACE="eth0"
VNET_BRIDGE="xenbr0"
VNET_MODE="STATIC"
VNET_SUBNET="192.168.X.X"
VNET_NETMASK="255.255.255.XX"
VNET_BROADCAST="192.168.XX.XX"
VNET_ROUTER="192.168.XX.XX"
VNET_DNS="XX.XX.XX.XX"
VNET_MACMAP="AA:DD:11:CE:FF:ED=192.168.XX.XX \
 AA:DD:11:CE:FF:EE=192.168.XX.3XX"
```

En el master iniciamos el servicio del CLC y del CC, y en cada unos de los nodos iniciamos el servicio del Node Cobntroller.

Configuración de Eucalyptus

Con los servicios levantados podemos empezar a definir la morfología de nuestra infraestructura, en primer lugar en necesario registrar el clúster, asociando el clúster al CLC y a su vez indicar cada uno de los nodos que forman parte del clúster.

```
# En el master
# Registrar cluster
euca_conf -cc Y -cloud Y -nc N /etc/eucalyptus/eucalyptus.conf
euca_conf -addcluster <clustername> <clusterhost>
# Resgistrar nodos
euca_conf -nodes "<hostname1> ... <hostnamen>"  \
/etc/eucalyptus/eucalyptus.conf
euca_conf -cc N -cloud N -nc Y -instances /usr/local/instances \
/etc/eucalyptus/eucalyptus.conf
# Propagar claves
euca_sync_key -c /etc/eucalyptus/eucalyptus.conf
```



(a) Configuración CLC y WS3 (b) Configuración CC

Figura 24: Configuración vía web

Estas tareas pueden realizarse mediante la interfaz web, donde se permite registrar nuevos clústers y definir el tipo de instancias que se ejecutaran sobre los nodos de ese clúster (small,large, extralarge).

Para poder utilizar las imágenes debemos almacenarlas en Walrus para que cada NC pueda disponer de ellas.

```
# Almacenamiento en WS3
# Bucket corresponde al directorio donde se almacenaran
# 1 Creamos el fichero manifest con las partes de la imagen
# 2 Cargamos esta informacion en WS3
# 3 Registramos la imagen para ser utilizada
# Kernel
ec2-bundle-image -i vmlinuz-XX --kernel true
ec2-upload-bundle -b kernel-bucket -m vmlinuz-XX.manifest.xml
ec2-register <kernel-bucket>/vmlinuz-XX.manifest.xml
# Sistema de ficheros raiz
ec2-bundle-image -i test.img
ec2-upload-bundle -b image-bucket -m test.img.manifest.xml
ec2-register image-bucket/test.img.manifest.xml
# Ramdisk (opcional)
ec2-bundle-image -i initrd.img --ramdisk true
ec2-upload-bundle -b ramdisk-bucket -m initrd.img.manifest.xml
ec2-register <bucket_name>/initrd.img.manifest.xml
```

Podemos asociar la imágenes entre si para asegurar un correcto depliegue de la VM.

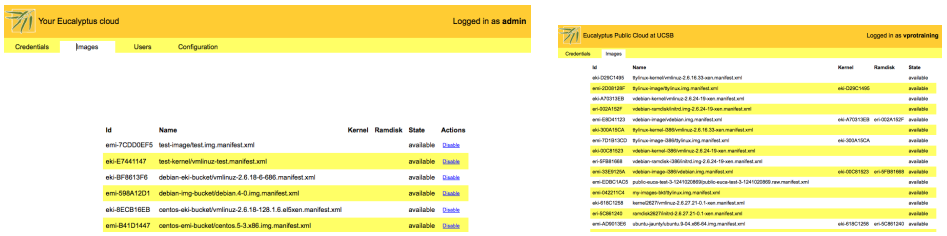
```
# Asociamos al crear el punto de almacenamiento
ec2-bundle-image -i test.img --kernel <eki-XXXXXXXX> \
--ramdisk <eri-XXXXXXXX>
# Asociamos al ejecutar la imagen ya registrada
ec2-run-instances <emi-XXXXXXXX> --kernel \
<eki-XXXXXXXX> --ramdisk <eri-XXXXXXXX>
```

Desde la interfaz web se puede predefinir un kernel y ramdisk por defecto,

a utilizar en el caso de no haber especificado ninguna relación. Esta acción se realiza desde el mismo punto donde anteriormente hemos podido configurar nuestro clúster y el tipo de instancias.

Es posible eliminarlas de WS3 cuando ya no sean de utilidad.

```
ec2-deregister <emi-XXXXXXXX>
ec2-delete-bundle -a EC2_ACCESS_KEY -s EC2_SECRET_KEY \
--url S3_URL -b <bucket> -p <file prefix>
```



(a) Imágenes WS3 (b) WS3 Eucalyptus Public Cloud

Figura 25: Imágenes Walrus

Interactuar con la infraestructura

Podemos decidir a que infraestructura realizamos una petición dependiendo de con que variables de entorno realizamos la petición. En las variables de entorno es donde se indican los servicios a los que realizar las peticiones y las credenciales que nos permitirán autentificarnos en estos servicios.

```
# Eucalyptus environment
EUCA_KEY_DIR=(dirname (readlink -f {BASH_SOURCE}))
export S3_URL=http://MASTER-HOST:8773/services/Walrus
export EC2_URL=http://MASTER-HOST:8773/services/Eucalyptus
export EC2_PRIVATE_KEY= \
{EUCA_KEY_DIR}/euca2-admin-XXXXXXXX-pk.pem
export EC2_CERT= \
{EUCA_KEY_DIR}/euca2-admin-XXXXXXXX-cert.pem
export EUCALYPTUS_CERT={EUCA_KEY_DIR}/cloud-cert.pem
export EC2_ACCESS_KEY='XXXXXXXXXXXXXXXXXXXXX'
export EC2_SECRET_KEY= \
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
```

En nuestro entorno de pruebas podremos acceder a tres infraestructuras, nuestra infraestructura interna, el cloud público proporcionado por el equipo de Eucalyptus (EPC), y el cloud público Amazon EC2.

Una vez autentificados ya somos capaces de desplegar las imágenes almacenadas en WS3 sobre nuestro clúster.

```
# Imagenes almacenadas
```

```
ec2-describe-images
# 1 CC + EBS Controller + nodes = availability zone
ec2-describe-availability-zones
# Clave para acceder a la instancia
ec2-add-keypair testkey > testkey.private
ec2-describe-keypairs
chmod 0600 testkey.private
# Instanciar la imagen
ec2-run-instances <emi-id> --kernel <eki-XXXXXXXX> \
--ramdisk <eri-XXXXXXXX> -k mykey -n <cantidad> -t <tipo>
# Informacion instancias
ec2-describe-instances
# Autorizar conexiones SSH
ec2-authorize group-name -P tcp -p 22 -s 0.0.0.0/0
# Conectar a la instancia
ssh -i testkey.private root@192.168.XX.XX
# Terminar instancia
ec2-terminate-instances id-instancia
```

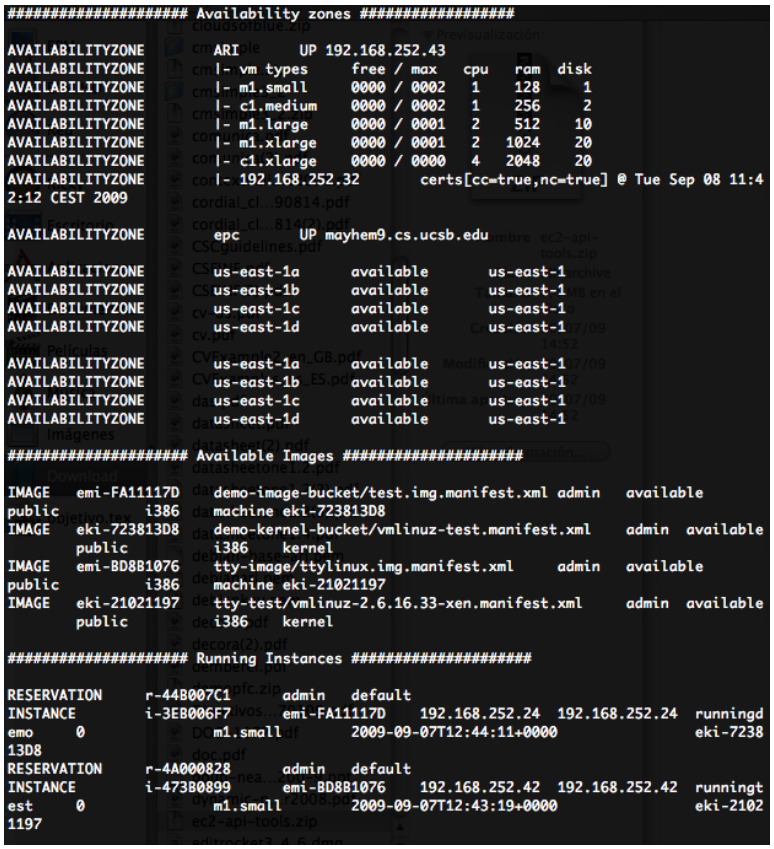


Figura 26: Entorno de trabajo Eucalyptus.

Prestaciones	Funciones
Interfaz de usuario	Proporciona una interfaz por línea de comandos (CLI) que permite gestionar el ciclo de vida de las VM y los recursos físicos. XML-RPC API y interfaz para libvirt que permite interactuar con la capa de virtualización
Planificador	Algoritmo Match-making. Políticas de asignación Paking Policy,Striping Policy, Load-aware Policy. Planificación avanzada mediante Haizea Lease Manager
Tecnologías de Vritualización	Soporta Xen, KVM y VMware en su última versión estable.
Gestión de imagenes	Driver que implementa un mecanismo de transferencia y clonación de imagenes.
Gestión de red	Definición de redes virtuales mediante un template de configuración que permiten definir las interconexiones entre VMs.
Compatibilidad EC2	Sí. Mediante la configuración de un driver se permite la ejecución de VM sobre EC2.
Tolerancia a fallos	Sistema de almacenamiento utilizando una BBDD persistente que guarda la información referente a los recursos físicos y las VM para su recuperación.
Instalación	Se realiza la instalación en una máquina que actúa como master de la infraestructura, no es necesaria la instalación de un nuevo servicio en los nodos. Distribuido en Ubuntu 9.04 desde Abril del 2009.
Licencia	Apache License, versión 2

Cuadro 5: Características del gestor OpenNebula

Prestaciones	Funciones
Interfaz de usuario	Vía web se gestionan las credenciales de autentificación, los usuarios, la configuracón del cluster y tipo de instancias, también se monitorizan las imágenes almacenadas en nuestro repositorio. Interfaz mediante linea de comandos (CLI), las mismas interfaces que utiliza Amazon (EC2-api-tools, EC2-ami-tools).
Planificador	Permite configurar la configuración política de planificación del CC. Existe dos tipos Greedy, RoundRobin.
Tecnología Vritualización	Xen, KVM.
Gestión de imagenes	Las herramientas de EC2 permiten interactuar con el repositorio de imágenes y aprovisionar cada VM a un nodo.
Gestión de red	Dispone de un subsistema de redes virtuales configurable pudiendo ser adaptado en diferentes entornos de red. Ofrece estos modos: SYSTEM MODE, STATIC MODE, MANAGED MODE, MANAGED-NOVLAN
Compatibilidad EC2	Sí. Se realiza de una forma directa ya que utiliza las mismas APIs para controlar la infraestructura. Únicamente es necesario definir las variables de entorno adecuadas para interactuar con cada una de las infraestructuras.
Tolerancia a fallos	Walrus (WS3) es un servicio de almacenamiento compatible con Amazon S3. Block Storage es un servicio copatible con Amazon Elastic Block Storage. Herramientas de terceros (s3curl, s3cmd, s3fs)
Instalación	Consiste en la instalación de tres componentes, el Cloud Controller (CLC), el Cluster Controller (CC) generalmente instalados en el master y interactuan con cada uno de los nodos donde se instala el Node Controller(NC) , las redes donde se instalan cada uno de los componentes deben poder comunicarse entre si. Ubuntu 9.04 lo distribuye como gestor recomendado formando el Ubuntu Enterprise Cloud.
Licencia	GPL, versión 3

Cuadro 6: Características del gestor Eucalyptus

9. Ejemplos

9.1. Amazon bajo demanda

LAMP en menos de 5 minutos

Este ejemplo muestra la capacidad de Amazon EC2, de prestar sus servicios bajo demanda. El coste asociado a la prestación de este servicio se fracciona por horas según el tipo de instancia, *small* en nuestro ejemplo.

El software sobre nuestro entorno de desarrollo ocasionalmente presenta unos requisitos específicos respecto al sistema donde se ejecuta (SO, versiones de programas instalados) y los recursos propios pueden no estar disponibles o no ajustarse a estos requisitos. Ante esta circunstancia, es necesario proveerse de nuevos recursos o adquirirlos mediante un proveedor externo.

Amazon EC2 permite no incurrir en la adquisición de nuevo hardware y consumir por hora la utilización de máquinas virtuales. Necesitamos el servicio de almacenamiento que proporciona Amazon para poder ejecutar nuestros propios entornos.

Utilizaremos una imagen base del repositorio público de Amazon, con el software ya instalado. Esto nos permite ahorrar el tiempo asociado a la creación de nuestra imagen, almacenamiento y registro. Esta imagen, contiene todo el software necesario para actuar como un servidor LAMP (Linux, Apache, MySql, Php).

9.1.1. Ejecución del ejemplo

- Crear una cuenta para disponer de los AWS: EC2, S3
- Acceder a alguna de la herramientas para gestionar nuestro WS
  - Amazon Magament Console
  - ElasticFox
  - CLI: EC2-api-tools, EC2-ami-tools
  - OpenNebula plugin
- Instanciar la imagen pública con el software ya instalado.
- Conectarse a instancia y realizar nuestras modificaciones especificas.
- Acceder desde un navegador a nuestra página, lo que aprovechamos para comunicar con las otras interfaces web de nuestro entorno de pruebas.

9.2. OpenNebula sobre Amazon

Configuraciones adicionales sobre el driver de OpenNebula para interactuar con Amazon.

```
# /etc/one/im_ec2/im_ec2.conf
# Capacidad maxima a utilizar de EC2
SMALL_INSTANCES=5
LARGE_INSTANCES=2
EXTRALARGE_INSTANCES=1
#
#/etc/one/vmm_ec2/vmm_ec2.conf
# Configuraciones de la VM sobre EC2
    EC2 = [ keypair = "testKEY",
                authorizedports = 22,
                instancetype = "m1.small"]
#
# /etc/one/vmm_ec2/vmm_ec2rc
# Configuracion de la cuenta AWS
#CLASSPATH=""
EC2_HOME="/opt/home/eucalyptus/tools/ec2-api-tools-1.3-30349"
EC2_PRIVATE_KEY=
"/.euca/pk-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.pem"
EC2_CERT=
"/.euca/cert-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.pem"
#
# /etc/one/oned.conf
# Configuracion driver
  IM_MAD = [
      name       = "im_ec2",
      executable = "one_im_ec2",
      arguments  = "im_ec2/im_ec2.conf",
      default    = "im_ec2/im_ec2.conf" ]
  VM_MAD = [
      name       = "vmm_ec2",
      executable = "one_vmm_ec2",
      default    = "vmm_ec2/vmm_ec2.conf",
      type       = "ec2" ]
```

Después de las configuraciones es necesario añadir a EC2 en la lista de nodos a los que el master puede aprovisionar.

```
onehost create ec2 im_ec2 vmm_ec2 tm_dummy
```

De la misma forma que con el resto de las VM de nuestro entorno, la AMI se despliega mediante *onevm* a partir de un template con las características de la instancia.

```
# EC2 template
EC2 = [ AMI="ami-xxxxxxxx",
    KEYPAIR="testKEY",
```

```
        ELASTICIP=”xx.xx.xx.xx”,
        AUTHORIZED_PORTS=”22”,
        INSTANCETYPE=m1.small]
    # Recurso a utilizar
    REQUIREMENTS = ’HOSTNAME = ”ec2”’
```

9.3. Provisionamiento Eucalyptus

Desde nuestro master podemos interactuar con 3 infraestructuras (recursos internos, Cloud Público Eucalyptus, Cloud Público Amazon). Este ejemplo muestra los pasos necesarios para el despliegue sobre Eucalyptus de un entorno de trabajo en nuestra infraestructura interna.

- Credenciales para la infraestructura interna.

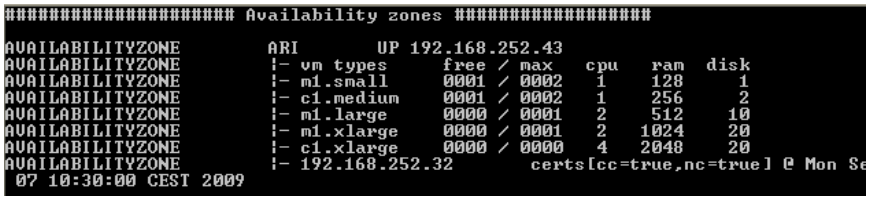


Figura 27: Recursos internos disponibles

- Almacenamiento del kernel de la imagen.

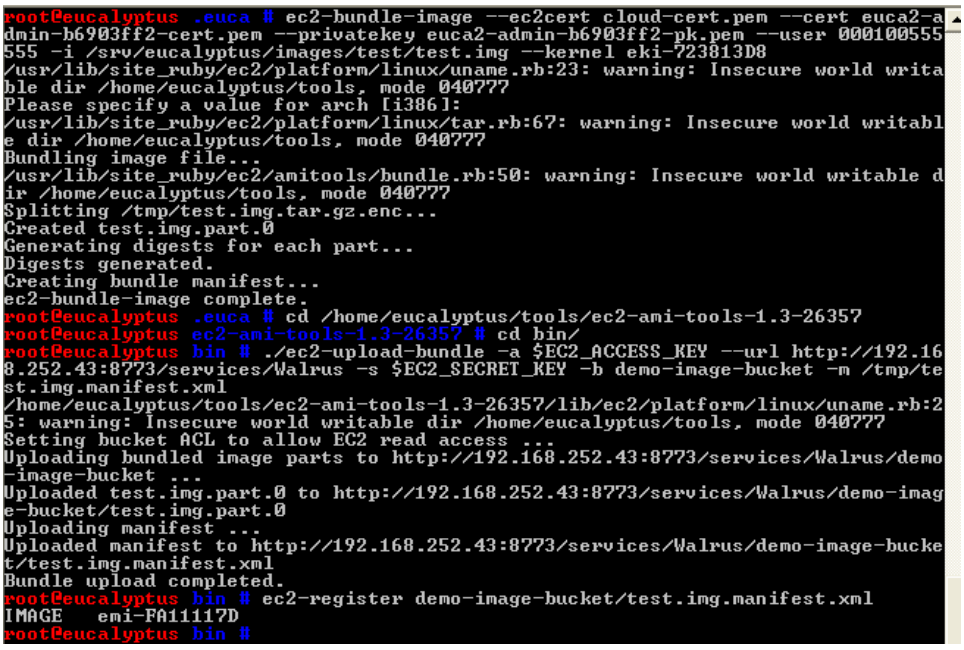


Figura 28: Alacenamiento núcleo y sistema de ficheros en WS3

- Almacenamiento del sistema de ficheros de la imagen.

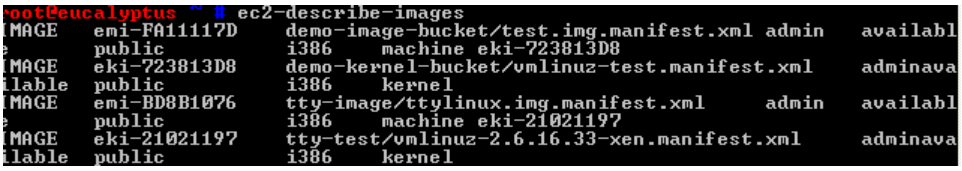


Figura 29: Imagenes disponibles en WS3

- Asociar el kernel al sistema de ficheros.
- Crear claves para conectarnos a la imagen

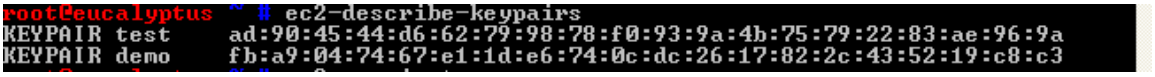


Figura 30: Claves de conexión

- Arrancar la imagen sobre nuestra infraestructura.

Figura 31: Ejecución de la instancia

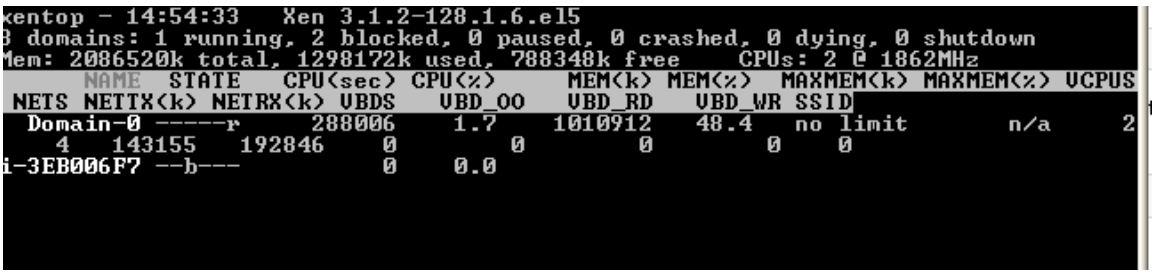


Figura 32: Monitorización VM

- Conectarnos a la instancia.

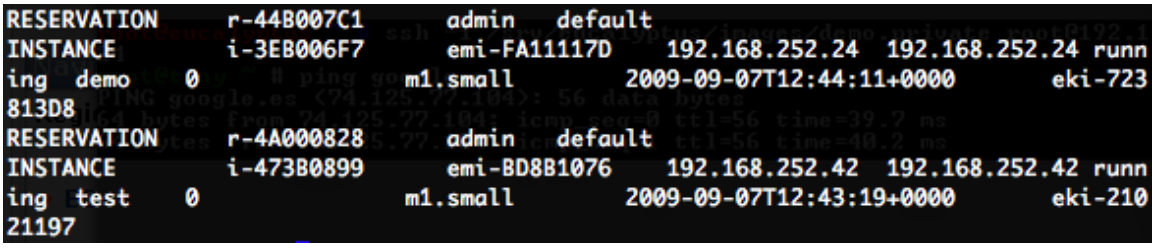


Figura 33: Instancias en ejecución

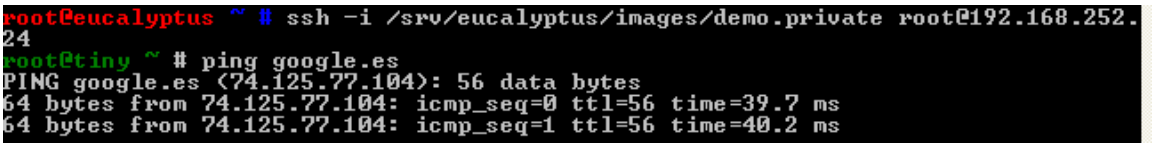


Figura 34: Conexión a la instancia

9.4. IaaS para los entornos Grid

El Grid Computing es una tecnología de computación distribuida, que pretende solventar situaciones de computo con grandes ciclos de procesamiento o gran número de datos a tratar. Mediante un *middleware* somos capaces de dividir un programa en partes que se ejecutan sobre un conjunto de ordenadores.

Atos Reseach & Innovation realizo un caso de uso en el marco del proyecto europeo GridCOMP. Con la finalidad de colaborar en el desarrollo del *middleware* Proactive, el caso de uso DSO (Days Sales Outstanding) divide un proceso pesado implementado en PL/SQL entre diversos nodos de trabajo, estos realizan los cálculos para obtener el resultado final.

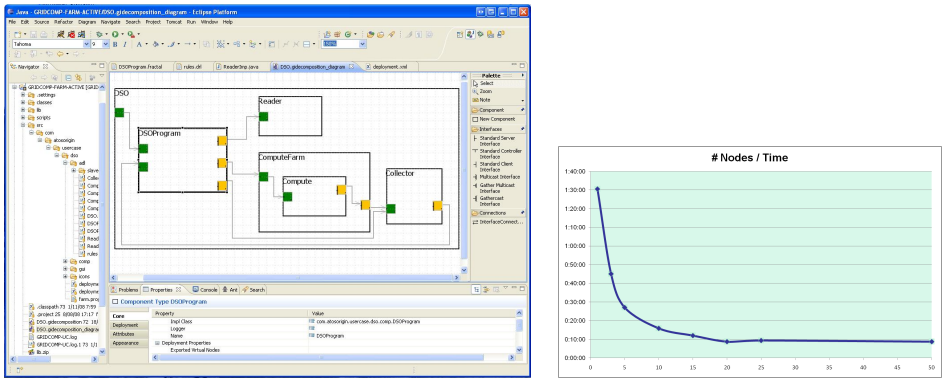
La virtualización permite encapsular los diferentes tipos de nodos que intervienen en nuestro entorno. Un master es el encargado de controlar la ejecución de la aplicación, divide el proceso en diferentes tareas y las envía a los nodos de trabajo. Estos nodos reciben estas tareas y las procesan, enviado los resultado de vuelta al nodo master.

Durante la implementación del prototipo se desplegó la aplicación grid sobre diferentes infraestructuras virtualizadas.

- Máquinas virtuales VMWare (Windows/Linux) en los recursos internos de la empresa.
- Sobre Grid5000 una infraestructura distribuida en 9 emplazamientos de Francia.

En el marco de este proyecto, la implantación de una aplicación grid sobre nuestra infraestructura, es una buena manera de probar nuestro entorno en

un caso de desarrollo de software real. Es necesario la encapsulación de los diferentes comportamientos de nuestra máquinas virtuales en una tecnología capaz de ser desplegada en nuestra infraestructura (Xen), se ha iniciado la creación de las VM mediante la herramienta *xen-tools*, no obstante este caso de uso no ha sido desplegado sobre nuestro entorno de pruebas.



(a) Componentes Proactive (b) Resultados sobre los nodos

Figura 35: Ejemplo GridCOMP

10. Resultados y conclusiones

El Cloud Computing esta adquiriendo gran relevancia como tecnología emergente, las grandes empresas del software mundial están ofreciendo sus propias soluciones para acceder a estos servicios prestados a través de Internet. La tendencia en los proyectos de investigación en Europa, anteriormente marcada por la investigación en el Grid Computing (figura 36) ha dejado paso a la investigación de las nuevas oportunidades que ofrece el modelo Cloud Computing y la virtulización.

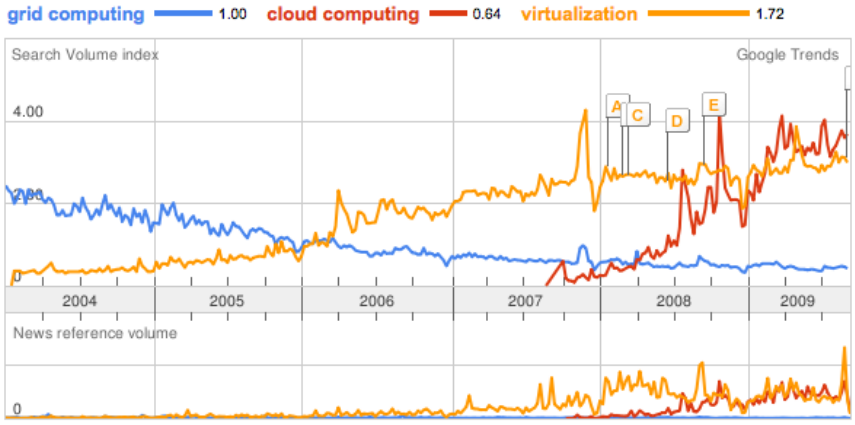


Figura 36: Resultado Google trends

La virtualización de nuestros recursos, ha permitido solventar los problemas de rigidez que sufría la infraestructura y desacoplar los entorno de trabajo para cada proyecto. La gestión eficiente de estos recursos, evita que los beneficios ofrecidos por la virtualización se vuelvan en contra. Un entorno virtualizado sin control es susceptible de presentar problemas de sobre-expansión, perdida de información. Una mala asignación de los recursos disminuye el rendimiento de la infraestructura y de nuestras aplicaciones.

La profundización en el estado del arte de los dos gestores implantados nos ha permitido evaluar las capacidades de estos. Este proyecto no pretende tomar una decisión en firme de cuál es el más adecuado para nuestra infraestructura, ya que esta toma de decisión se basará en la funcionalidad y características finales del servicio.

- **Eucalyptus,** plantea la organización de nuestra infraestructura en clústers, formados por nodos de trabajo, esto permite separar los clústers según el propósito final de los procesos ejecutados en cada clúster. La planificación es evaluada a nivel de clúster, este a su vez es capaz de aplicar una planificación entre sus nodos. La configuración de red se realiza de forma genérica en el Cloud Controller y toda la infraestructura actuará en consecuencia con esta configuración. El gestor Eucaly-

ptus esta enfocado a la prestación de infraestructuras como un servicio, el funcionamiento del gestor esta fuertemente ligado al comportamiento que ofrece Amazon EC2, este hecho nos beneficia si nuestro servicio pretende trabajar de la misma manera que lo hace EC2.

- **Open Nebula,** ejerce como gestor central de una infraestructura formada por nodos físicos en los que poder proveer máquinas virtuales, la encapsulación de diferentes propósitos se realiza a nivel de red . La planificación es aplicada por el master de la infraestructura, para cada VM podemos definir que política de planificación se aplica. La configuración de red se realiza a nivel de máquina virtual por medio de la configuración de *templates* específicos para ello (asignación de una dirección en concreto o descripción de un rango de direcciones disponibles). Open Nebula esta diseñado de una manera menos rígida que Eucalyptus, permitiendo implementar diferentes escenarios y/o servicios a ofrecer.

El entorno de pruebas implementado durante el proyecto servirá a Atos Research & Innovation para reaprovechar los ordenadores excedentes que actualmente están en desuso. La implementación del gestor Open Nebula se reaprovechará en el marco del proyecto Europeo NUBA donde Atos Origin colaborá y donde OpenNebula es uno de los gestores con los que interactuar.

10.1. Líneas futuras

- La fase 3 (gestión de la infraestructura) requiere de funcionalidades que no han sido implementadas en este proyecto.
  - Backup eficiente de los recursos y los datos en la infraestructura.
  - Gestión de usuarios
  - Automatizar la creación de los recursos virtuales.
- Tras el proyecto somos capaces de abordar la fase 4 y 5 fijadas para la adopción.
- El entorno de pruebas puede ser redefinido para realizar tests sobre las temáticas bajo investigación en el Cloud Computing (estándares, pruebas de interoperabilidad, nuevas políticas de planificación, etc).
- Comprobar el funcionamiento del gestor con otras tecnologías de virtualización.

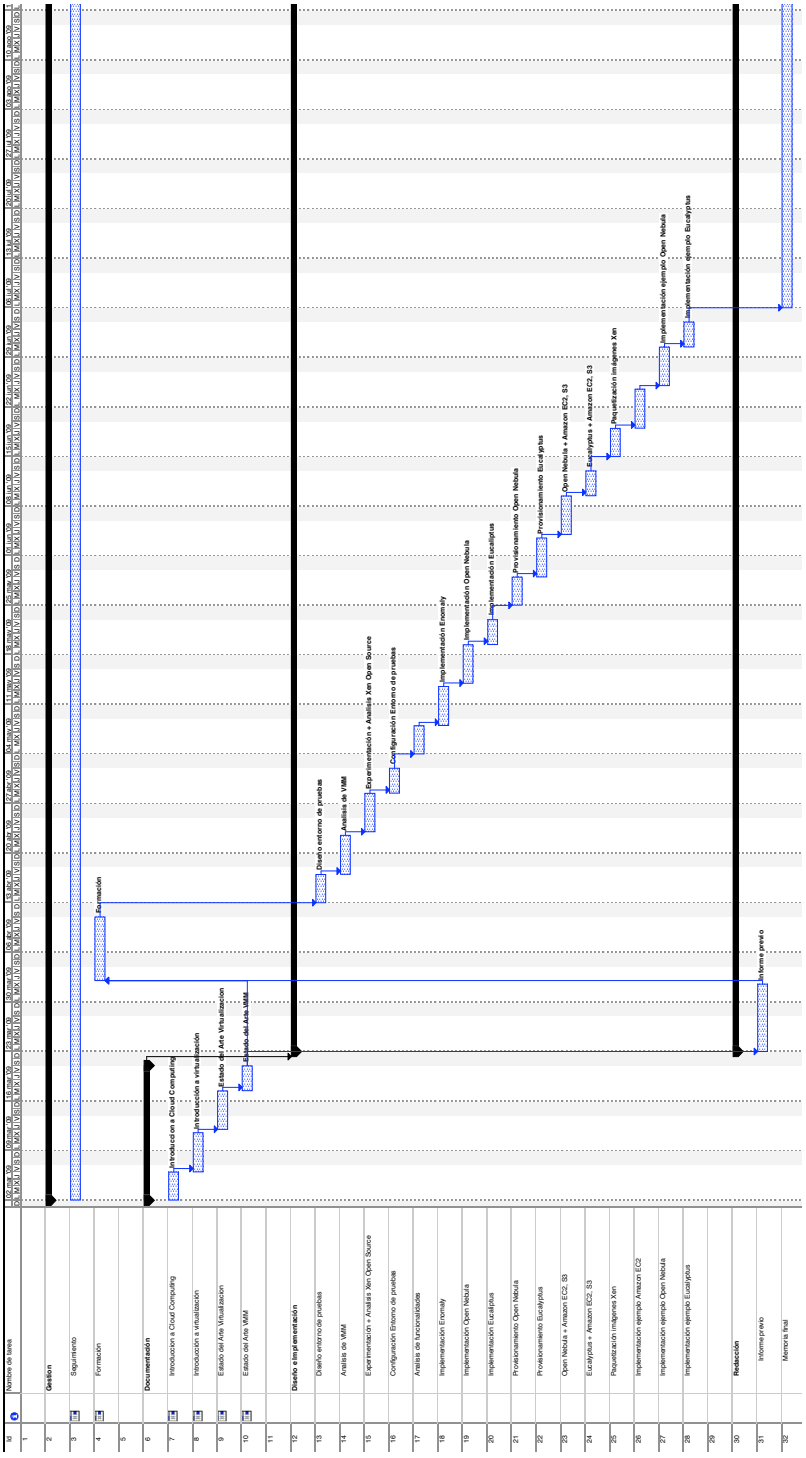
Anexos

A. Glosario

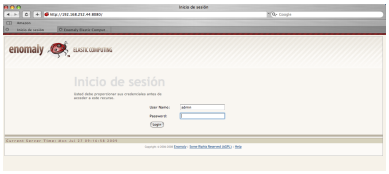
- **Cloud Computing,** es una tecnología que permite la prestación de servicios de computación a través de Internet.
- **Cloud,** utilizado para referirse a la infraestructura que permite la prestación del servicio y habitualmente puede ser sinónimo de Internet o una zona de Internet utilizada por la infraestructura.
- **Utility Computing,** es la capacidad de ser provisto exclusivamente de los recursos esenciales para mi proceso de negocio. Al igual que otros servicios tradicionales (agua,gas,electricidad) el utility computing ofrece bajo demanda servicios computacionales como procesamiento y almacenamiento.
- **Grid Computing,** ofrece la capacidad de distribuir procesos entre diferentes recursos con la finalidad de resolver un problema dividiendo la ejecución del problema entre diferentes ordenadores.
- **Hipervisor,** es el monitor de las máquinas virtuales también denominado Virtual Machine Monitor (VMM). Es el software encargado de realizar la virtualización del sistema y de garantizar su funcionamiento.
- **Service Oriented Architecture (SOA),** infraestructura o aplicaciones formadas por una colección de servicios capaces de comunicarse y coordinar tareas en el sistema.
- **Web Services (WS),** es un sistema de software que permite la conexión entre las arquitecturas orientadas a servicio. Proporciona interoperabilidad en la interacción entre máquinas sobre la red. Web Service Description Language (WSDL) es una lenguaje basado en XML que permite describir los WS.
- **Software as a Service (SaaS),** es un modelo de distribución de software donde los proveedores asumen el mantenimiento de la aplicación y brindan el servicio final al consumidor (generalmente vía web) sin necesidad de instalación de software ni mantenimiento.
- **Platform as a Service (PaaS),** es la entrega de una plataforma de computación capaz de interactuar mediante entornos de desarrollo y APIs bien definidas con los servicios del sistema.
- **Infrastructure as a Service (IaaS),** es la entrega de una infraestructura de computación como un servicio. Generalmente se entrega en forma de máquina virtual.
- **Everything as a Service (XaaS),** este concepto contempla la capacidad de ofrecer todo el stack de nuestra infraestructura mediante la interconexión de servicios.

- **Web 2.0,** es como se define a la nueva generación en el desarrollo de la tecnología web, caracterizada por una nueva gama de servicios para interactuar con la tecnología así como la utilización que hacen los usuarios de esta tecnología (comunidades, blogs, wikis).
- **Service Level Agreement (SLA),** es una contrato donde se definen los términos del servicio.
- **CAPEX,** se refiere a gastos de capital realizados por las empresas con la finalidad de adquirir un activo o añadir valor a un activo ya adquirido.
- **OPEX,** se refiere a gastos de funcionamiento derivados de la utilización de un producto o de la realización de una tarea.
- **Representational State Transfer-REST,** es una estilo de arquitectura de software para sistemas hipermedia distribuidos como el World Wide Web.
- **Simple Access Object Protocol-SOAP,** es un protocolo de comunicación diseñado para intercambiar mensajes en formato XML entre una red de ordenadores, normalmente sobre HTTP.
- **Network Bridge,** es un dispositivo de interconexiónd de redes que conecta dos segmentos de red como una sola red, usando el mismo protocolo de establecimiento de red.
- **XML-RPC,** especificaciones que permiten al software ejecutarse entre diversos sistemas operativos. Realiza llamadas remotas a procedimientos a través de Internet, usando HTTP como transporte y XML como codificación.
- **Centro Procesamiento de Datos (CPD),** ubicación donde se encuentran los recursos computacionales para el procesamiento de la información.
- **DMZ,** o zona desmilitarizada es una subred de una LAN situada entre la red privada en una organización ya la red externa.
- **Servidor LAMP,** es un cojunto de subsistemas software (Linux,Apache, MySql, PHP) que conforman un entorno donde poder desarrollar aplicaciones web.

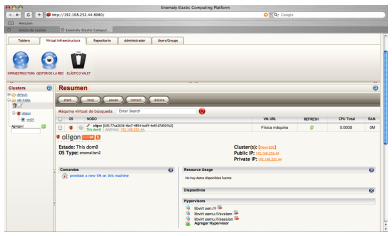
B. Planificación del proyecto



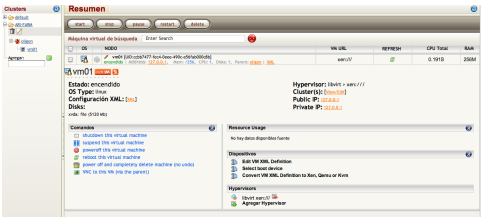
C. El gestor Enomaly



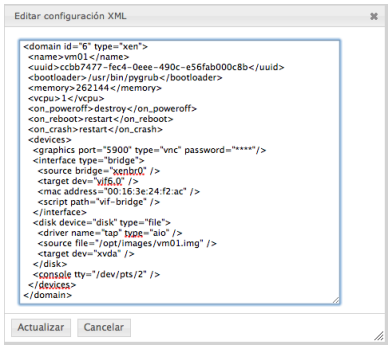
(a) Login



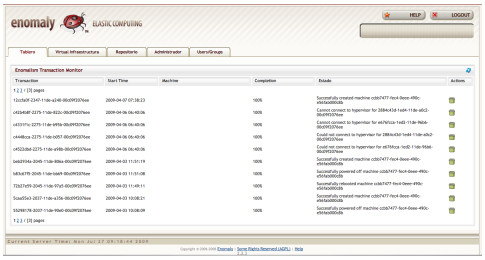
(b) Máquina física



(c) Máquina virtual

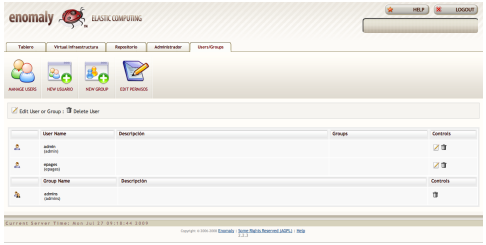


(d) Definición máquina virtual



(e) Transacciones

(f) Repositorio



(g) Gestión de usuarios

Figura 37: Imágenes Enomaly

Prestaciones	Funciones
Interfaz de usuario	Ofrece un panel de control vía web como punto central de la gestión de la infraestructura.
Planificador	El administrador es capaz de planificar el lanzamiento de VM sobre los nodos a través del panel de control, mediante la utilidad *Elastic Valet*.
Tecnología Vritualización	Xen, KVMy el conector libvirt.
Gestión de imagenes	Existe una herramienta de creación de VM con soporte a virtualización por hardware. Somos capaces de gestionar diferentes repositorios de VM que pueden ser desplegadas sobre un clúster en concreto.
Gestión de red	Posibilidad mediante web de definir una rango válido de direcciones sobre un clúster en concreto.
Compatibilidad EC2	No existe documentación para la versión abierta.
Tolerancia a fallos	Base de datos MySql en cada uno de los ordenadores para almacenar toda la información.
Instalación	Se instala ECP en cada uno de los nodos, no hay una relazión master/slave entre ellos. Se instala una base de datos MySqlp y un repositorio de ficheros mediante NFS que es compartido por toda la infraestructura. Los nodos se comunican utilizando llamadas REST ente ellos.
Licencia	Affero GNU Public License.

Cuadro 7: Características del gestor Enomaly

Referencias

[1] Nickolai Kondratieff. The major economic cycles, 1925.

[2] *4th International Conference, ICETE 2007, Barcelona, Spain, July 28-31, 2007, Revised Selected Papers*, volume 23. Springer Berlin Heidelberg, November 2008.

[3] Stefanie Olsen. Sun's john gage joins al gore in clean-tech investing. *CNET*, 2008.

[4] Forrester. The new tech ecosystems of cloud, cloud services, and cloud. *Future View*, 2008.

[5] Gartner. Defining and describing an emerging phenomenon, cloud computing. *paper*, 2008.

[6] Telefonica I+D. A break in the clouds: Towardsa cloud definition. *CloudScape*, 2008.

[7] Jeremy Geelan. Twenty-one experts define cloud computing. *Virtualiztion Journal*, 2009.

[8] Dmitry Sotnikov. Cloud definitions: Nis gartner forrester. *CloudEnterprise.info*, 2009.

[9] Peter Mell and Tim Grance. Draft nist working definition of cloud computing. *National Institue of Standars and Technology*, 2009.

[10] HP. The next wave: Everything as a service. *Executive Viewpoint*, 2008.

[11] SalesForce. http://www.salesforce.com/saas/.

[12] Google. http://code.google.com/intl/ca/appengine.

[13] Microsoft. http://www.microsoft.com/azure.

[14] Amazon. http://aws.amazon.com.

[15] Songnian Zhou. Adaptation image of clouds moving into the enterprise. *Platform Computing*, 2009.

[16] Resources and services virtualization without barries. *project*, 2008.

[17] Gerald J. Popek and Robert P. Golberg. Formal requeriments for virtualizable third generation architectures. *paper*, 1974.

[18] Gartner Inc. Gartner estimates ict industry accounts for 2 percent of global co2 emissions (nota de prensa). *paper*, Abril 2007.

[19] Bob Worrall. Reducción de costes y calor en el centro de datos. *Inner Circle Sun.com*, 2008.

[20] Sun Microsystems. Take your business to a higher lever. *Sun.com*, 2009.

[21] rPath. The pragmatist's guide to cloud computing. *white paper*, 2008.

[22] Amazon.com. Amazon news release 2007. *Amazon News*, 2007.

[23] William Von Hagen. *Professional Xen Virtualization.* Wrox Professional guides, 2008.

[24] Xen. Xen architecture overview. *paper*, 2008.

[25] B. Sotomayor R. S. Montero I. M. Llorente and I. Foster. Virtual infrastructure management in private and hybrid clouds. *IEEE Internet Computing, in press*, 2009.

[26] Daniel Nurmi Rich Wolski Chris Grzegorczyk Graziano Obertelli Sunil Soman Lamia Youseff Dmitrii Zagorodnov. The eucalyptus open-source cloud-computing system. *9th IEEE International Symposium on Cluster Computing and the Grid, Shanghai, China*, 2008.

Enric Pagès Montanera
18 de septiembre de 2009

Este proyecto tiene como finalidad ofrecer un servicio de computación en forma de máquina virtual, utilizando los recursos internos de Atos Research & Innovation. Además, se pretende implementar este servicio sobre los excendentes de máquinas del propio departamento. La prestación de este servicio se realiza mediante un gestor de la infraestructura de forma centralizada. Para la implantación de este entorno se ha definido la adopción en fases y profundizado en dos de los gestores más activos en la investigación del modelo Cloud Computing (Open Nebula, Eucalyptus).

This project aims to offer a computational service as a virtual machine, using the Atos Research and Innovation's internal resources. Furthermore, it aims to implement this service upon the surplus machines of this department. The delivery of this service will be carried out using a centralised management infrastructure. To achieve this end, adoption according to phases has been defined and in depth investigation of the two management tools most used in cloud computing reserch has been carried out (Open Nebula, Eucalyptus).

Aquest projecte te com a finalitat oferir un servei computacional en forma de máquina vitual, utilitzant els recursos interns de Atos Reseach & Innovation. A més, es preten implementar aquest servei sobre les màquines excedents d'aquest departament. La prestació del servei es realitza mitjantçant un gestor per a la infraestructura de forma centralitzada. Per a la implantació d'aquest entorn s'ha definit l'adopció en fases y profunditzat en dos dels gestors més actius en l'investigació del módel Cloud Computing (Open Nebula, Eucalyptus).