Proyecto Fin de Máster en Ingeniería de Computadores

Curso 2008/2009

Ejecución de una base de datos distribuida sobre un entorno de Cloud Computing

Autor:
Pablo Martos Rodríguez

Dirigido por:
Dtor. José Ignacio Martín Llorente
Dpto. Arquitectura de Computadores y Automática

Colaboración externa de dirección:
Manuel Couceiro Domínguez (Ericsson España S.A.)

Máster en Investigación en Informática

Facultad de Informática

Universidad Complutense de Madrid

Autorización

El abajo firmante, matriculado en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: "Ejecución de una base de datos distribuida sobre un entorno de Cloud Computing", realizado durante el curso académico 2008-2009 bajo la dirección de Ignacio Martín Llorente y con la colaboración externa de dirección de Manuel Couceiro Domínguez (Ericsson España S.A.) en el Departamento de Arquitectura de Computadores y Automática, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Firmado:

Pablo Martos Rodríguez

Agradecimientos

En primer lugar, me gustaría dar las gracias a Ericsson España S.A. por ofrecerme desarrollar este proyecto con ellos y por la experiencia adquirida durante la realización del mismo. Más concretamente, me gustaría agradecer la ayuda recibida por parte del departamento de Research y de su director Miguel Ángel Pallares. En especial, me gustaria dar las gracias a Manuel Couceiro, quien ha sido mi tutor durante estos meses y a Javier Torres. Ambos me han proporcionado una valiosa ayuda y me han aportado numerosos conocimientos, gracias a los cuales he sido capaz de conseguir alcanzar mis objetivos satisfactoriamente.

Por último, doy las gracias a mis familiares y amigos, muy especialmente a mis padres, cuyo apoyo constante ha sido muy valioso para continuar.

Índice General

Índice de imágenes

Resumen del proyecto

Resumen

Este proyecto se centra en el paradigma de computación distribuida denominado Cloud Computing. En el primer capítulo se proporciona una visión general sobre este modelo. En primer lugar, se intenta clarificar el término, debido a la gran confusión existente en torno a él. A continuación se muestran los diferentes tipos en los que se divide y se diferencia entre los conceptos de nube pública, privada e híbrida. El capítulo termina con una comparativa entre grids y clouds y finalmente se enumeran las distintas ventajas e inconvenientes que ofrece.

El segundo capítulo está dedicado al estado del arte. En él, se muestran algunos de los principales servicios y herramientas existentes en la actualidad. Además, se indican las empresas líderes del sector y se ha creado un modelo de referencia sobre la arquitectura general del ecosistema del Cloud Computing.

En el tercer capítulo se presenta una comparativa entre algunos de los servicios más destacados.

Seguidamente, en el cuarto capítulo se describe la arquitectura tipo cloud construída y la base de datos (BD) ejecutada sobre ella. También se han listado una serie de aspectos relacionados con la gestión de datos por parte de la BD (data management) que pueden verse afectados por arquitecturas tipo cloud.

La última parte del proyecto se resume en el quinto capítulo. Aquí se exponen los resultados obtenidos al comparar la ejecución de la base de datos distribuida sobre distintos entornos: hardware tradicional, máquina virtual y arquitectura cloud y las conclusiones extraídas.

Palabras clave

Computación en nube, virtualización, computación distribuida, base de datos, escalabilidad, código libre.

Abstract

This project focuses on a distributed computing paradigm named Cloud Computing. In the first chapter an overview of this model is provided. First of all, it is sought to clarify the term, due to the great confusion around it. Besides, different types in which it is divided are shown, and the concepts of public cloud, private cloud and hybrid cloud are differentiated. The chapter ends with a comparison between grids and clouds and finally, they are listed the different advantages and disadvantages that it offers.

The second chapter is devoted to state of the art. Some of the main services and tools currently available are shown on it. In addition, sector leader companies are listed, and a reference model about the overall architecture of the Cloud Computing ecosystem has been created.

In the third chapter a comparison between some of the most distinguised services is presented.

Then, in the fourth chapter the built cloud architecture and the database executed on it are described. A number of aspects related to the data management that could be affected by cloud architectures have been listed as well.

The last part of this project is summarized in the fifth chapter. Here is presented the outcome for the comparison between the performance of the distributed database on different environments: traditional hardware, virtual machines and cloud architecture, and learnt conclusions are provided.

Key words

Cloud Computing, virtualization, distribuited computing, database, scalability, opensource.

1 Introducción

Esta sección ofrece una primera toma de contacto con el proyecto, explicando de la manera más simple y concisa posible los primeros detalles generales que se han considerado de interés.

1.1 Objetivos

Con este proyecto, nos introducimos en el mundo del Cloud Computing, muy de moda en la actualidad. Mucho se ha escrito y hablado acerca de este modelo de computación cada vez más extendido en nuestra sociedad y sin embargo, parece que el término aún sigue resultando confuso. Uno de los principales objetivos perseguidos aquí es precisamente proporcionar una visión general y clara acerca de este paradigma. El segundo objetivo consiste en proporcionar unos resultados concluyentes sobre la idoneidad de este tipo de arquitecturas para la ejecución de bases de datos (BDs), ya que es un tema que preocupa a varias empresas cuyo foco de negocio está relacionado con la gestión de datos o "data management". Para ello, se montó una arquitectura cloud, donde se desplegó una BD distribuida, y se compararon los resultados con los obtenidos al ejecutar esa misma BD en un entorno tradicional y en un entorno virtual.

En la actualidad, no existe apenas literatura relacionada con las intenciones de este proyecto, ya que los textos sobre este tema son todavía muy escasos. La mayoría de la información se encuentra en la web, donde no nos podemos quejar del número de páginas y blogs dedicados al tema en cuestión, y en artículos realizados por universidades y empresas. Es por ello, que este trabajo puede servir como punto de partida y orientación a quienes se estén planteando utilizar este tipo de arquitecturas en sus entornos.

1.2 Conceptos iniciales

En este apartado, se incluyen una serie de términos relacionados con el Cloud Computing que son de obligado conocimiento y que facilitarán el seguimiento del texto de aquí en adelante.

- **Virtualización:**

  Virtualización es la creación de una versión virtual de un sistema operativo, un servidor, un dispositivo de almacenamiento o un recurso de red. Consiste en la utilización de un software para posibilitar que un recurso físico (hardware) pueda ejecutar múltiples máquinas virtuales aisladas, con sus correspondientes sistemas operativos, simultáneamente. Tales máquinas virtuales tienen un comportamiento similar a máquinas reales independientes.

- **Hypervisor:**

  Un hypervisor, también llamado Virtual Machine Monitor (VMM), es un programa que permite que múltiples sistemas operativos compartan un único hardware de forma aislada y al mismo tiempo. Se comunica con el sistema anfitrión, ya sea hardware o sistema operativo, para interoperar entre las máquinas virtuales y el sistema físico. Como ejemplo de hypervisores tenemos VMWare, Xen, KVM, VirtualBox, etc.

- **Grid Computing:**

  Es la interconexión de varios ordenadores de diferentes dominios de administración para la obtención de computación de altas prestaciones. Según Ian Foster (*What is the Grid? A Three Point Checklist, 2002*), un Grid es un sistema que…

  1) …coordina recursos que no están sujetos a un control centralizado…
  2) …usando protocolos e interfaces estándar, abiertos y de propósito general…
  3) … para proporcionar calidades de servicio no triviales.

- **On-demand Computing**:

  Modelo de negocio de popularidad creciente en el que los recursos informáticos son proporcionados según los va necesitando el usuario (bajo demanda). Los recursos pueden estar dentro de la propiedad del usuario o su empresa, o ser proporcionados por un proveedor de servicios.

- **Elastic Computing:**

  Modelo de provisión de computación donde los recursos crecen o se reducen cuando es necesario (elástico), o lo que es lo mismo, la posibilidad de un sistema de adquirir o liberar recursos de computación bajo demanda.

- **Utility Computing:**

  Modelo de provisión de servicios informáticos que consiste en que el proveedor del servicio pone los recursos de computación y la administración de la infraestructura a disposición del usuario según los vaya necesitando y le cobra sólo lo que utilice (pago por uso) de forma similar al modelo utilizado por las compañías eléctricas.

Este modelo intenta maximizar la eficiencia en el uso de los recursos y/o minimizar los costes asociados.

- **Multi Tenancy:**

  Este concepto es uno de los elementos centrales de las aplicaciones cloud. Una arquitectura de aplicación multi-tenant es aquella donde todos los usuarios de la aplicación, independientemente de la organización a la que pertenecen, comparten la misma instancia del código, de la base de datos y de la infraestructura.

1.3 ¿Qué es Cloud Computing?

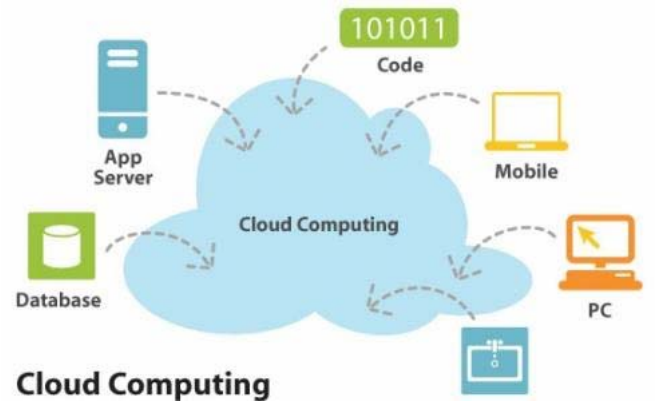

Figura 1 Cloud Computing

1.3.1 Definiciones

En la actualidad, el término Cloud Computing ("Computación en nube") está muy extendido y es utilizado de diversas formas y para diversos propósitos. A veces, incluso se ve influenciado por el marketing para dar valor a un producto.

Según un informe de la consultora McKinsey&Company, las nubes son servicios basados en hardware que ofrecen capacidades de computación, redes y almacenamiento de modo que:

- La gestión del hardware se abstrae mucho del comprador.
- Los compradores incurren en gastos de infraestructura como OPEX (herramienta para el cálculo de gastos de operación) variable por su utilización.
- La capacidad de la infraestructura es altamente elástica.

Para el IEEE Internet Computing, sin embargo, Cloud Computing es un paradigma en el que la información es almacenada en servidores de internet de forma permanente y cacheada temporalmente sobre los clientes (ordenadores de sobremesa, centros de entretenimiento, portátiles, etc.).

Si preguntamos a distintas personas, aún cuando estén relacionadas con el mundo de la informática, seguramente obtendremos tantas respuestas como número de interrogados. Incluso podemos encontrar respuestas en contra de este modelo:

> *Lo interesente sobre Cloud Computing es que lo hemos redefinido para incluir todo lo que ya hacemos … No entiendo que haríamos de forma diferente con el Cloud Computing, aparte de cambiar algunas palabras en nuestros anuncios.*
>
> Larry Ellison (CEO de Oracle), 26 de septiembre de 2008

> *Mucha gente se está subiendo al carro del cloud, pero no he escuchado a dos personas decir lo mismo sobre esto. Hay múltiples definiciones de "la nube".*
>
> Andy Isherwood (Vicepresidente de ventas de HP en Europa), 11 de diciembre de 2008

> *Es tan malo como usar software propietario. Haz tus tareas en tu propio ordenador y mantén una copia en un programa libre. Si utilizas un software propietario o el servidor web de otra persona estás sin defensas. Estás en las manos de la persona que desarrolló ese software. Es estúpido. Es peor que estúpido: es una gran campaña de marketing.*
>
> Richard Stallman (Free Software Foundation), 29 de septiembre de 2008

En el Cloud Computing Journal publicaron un artículo donde 21 expertos definieron el término. A continuación, se encuentran algunas de estas definiciones:

> *¿Qué es Cloud Computing? Amazon ha acuñado la palabra "elasticidad" que da una buena idea sobre las características clave de este paradigma: puedes escalar tu infraestructura bajo demanda en minutos, incluso hasta en segundos, en vez de días o semanas. Además, puedes evitar la inflautilización y sobreutilización de recursos. Con monitorización e incremento de provisión de recursos automático, pudiéramos despertarnos un día en un mundo donde no tengamos que preocuparnos sobre el escalado de nuestras aplicaciones web, porque ellas puedan hacerlo solas.*
>
> Markus Klems (http://markusklems.wordpress.com/)

> *Yo veo el Cloud Computing como una amplia lista de servicios basados en la web, con el objetivo de permitir a los usuarios obtener un amplio rango de funcionalidades mediante pago por uso. Anteriormente, se requerían tremendas inversiones en software y hardware, además de profesionales cualificados para adquirirlo. Cloud Computing es la puesta en práctica del modelo de utility computing sin complejidades técnicas ni preocupaciones.*
>
> Jeff Kaplan (http://www.thinkstrategies.com/blog/)

> *El modelo cloud se ha centrado inicialmente en hacer la capa de hardware accesible al usuario como computación bajo demanda y capacidad de almacenamiento. Es un primer paso importante, pero para que las compañías aprovechen toda la potencia del cloud, la infraestructura completa necesita escalar dinámicamente y ser fácilmente configurada, desplegada y gestionada bajo estos entornos virtualizados.*
>
> Kirill Sheynkman (Presidente y CEO de Elastra)

> *Cloud Computing es… la versión amigable de Grid Computing.*
>
> Trevor Doerksen (CEO y fundador de MoboVivo)

> *Cloud Computing se entiende realmente cuando se piensa en lo que necesitan los departamentos de IT: un camino para incrementar su capacidad o añadir capacidades "al vuelo", sin invertir en nueva infraestructura, ni formar a nuevo personal, ni comprar nuevas licencias de software. Cloud Computing abarca cualquier servicio de pago por uso en tiempo real, ofrecido sobre internet y que extiende las capacidades IT.*
>
> Bill Martin

…

Para evitar esta confusión, a partir de las definiciones anteriores y después de un exhaustivo estudio sobre este nuevo paradigma, podemos concluir que:

> Cloud Computing es un paradigma informático que consiste en que los recursos de computación son proporcionados como servicio ("as a service"), permitiendo a los usuarios acceder a servicios tecnológicos desde internet ("en la nube") bajo demanda.

Una tecnología clave para el surgimiento del Cloud Computing es la virtualización. No es necesaria, pero ayuda bastante. El Cloud Computing puede aprovechar al máximo los recursos al ser capaz de gestionar los entornos virtualizados bajo demanda. Esta es una característica que lo diferencia respecto al modelo de computación tradicional, ya que los entornos Cloud son escalables y capaces de ajustarse a las fluctuaciones en la demanda.

Construir tu propia infraestructura preparada para soportar picos en la demanda requiere grandes inversiones, además de espacio. Sin embargo, por muy grande que sea la infraestructura construida, puede no ser suficiente en un momento dado. Tener que elegir entre sobre-provisión e infra-provisión es un dilema clásico. En la figura siguiente, se puede observar este comportamiento:

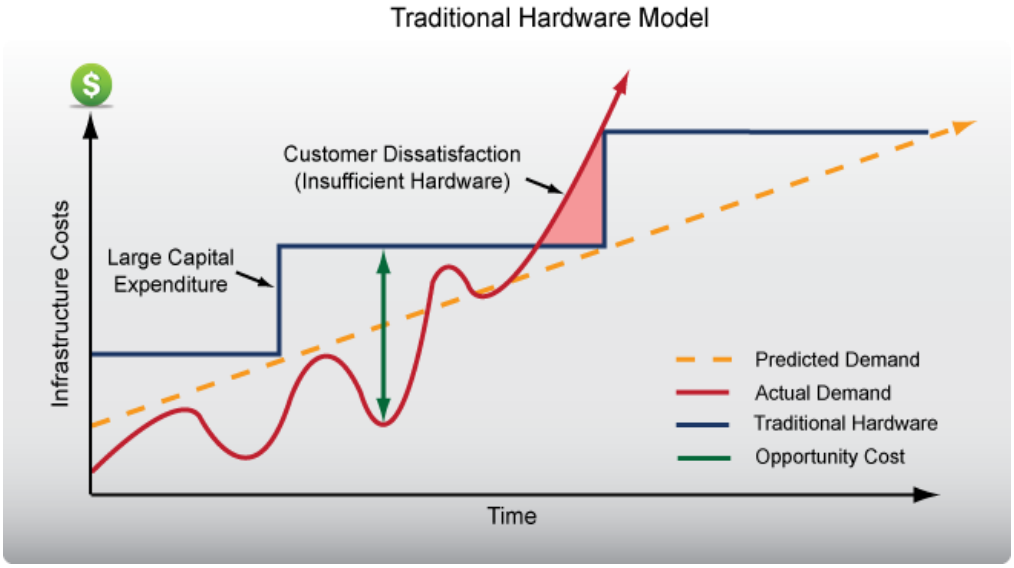

Figura 2 Modelo tradicional de computación (Fuente: www.rightscale.com)

Utilizando una infraestructura Cloud escalable es posible evitar este dilema. Este tipo de entornos reducen los costes iniciales y están preparados para ser capaces de responder satisfactoriamente a demandas pico en momentos puntuales. La figura siguiente refleja este comportamiento:

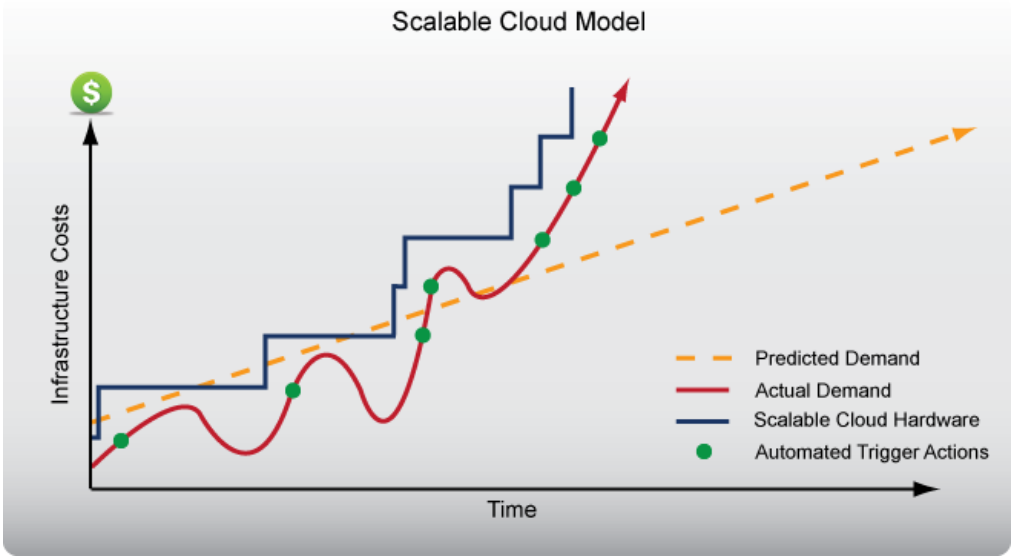

Figura 3 Modelo Cloud de computación (Fuente: www.rightscale.com)

1.3.2 Clasificación en capas

Como se ha comentado en el apartado anterior, Cloud Computing es un concepto muy general y que es utilizado de diversas formas. Sin embargo, parece que hay consenso con respecto a los tres tipos o capas fundamentales: Software como Servicio o Software as a Service (SaaS), Plataforma como Servicio o Platform as a Service (PaaS) e Infraestructura como Servicio o Infrastructure as a Service (IaaS).

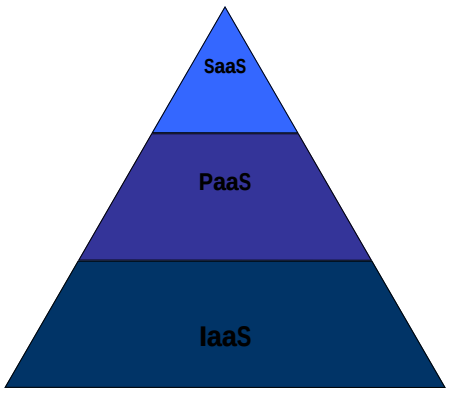

Figura 4 División en capas en Cloud Computing

- Software como servicio:

  Modelo de distribución de software en el que una aplicación es ofrecida como un servicio proporcionado a través de internet. Permite ofrecer una aplicación final dentro de una infraestructura bajo demanda, totalmente escalable tanto en número de usuarios como en requisitos de almacenamiento. En vez de instalar y mantener el software, simplemente se accede a él a través de la red, liberando a los clientes de la compleja administración de los sistemas. La empresa que aloja el software es la encargada de mantener la información del cliente, además de proveer los recursos necesarios para explotar dicha información.

- Plataforma como servicio (PaaS):

  El Cloud Computing ha cambiado drásticamente la forma en que las aplicaciones son construidas y ejecutadas. Se ha evolucionado a un modelo donde se ofrecen plataformas para construcción y ejecución de aplicaciones personalizadas, un concepto conocido como "Plataforma como servicio". Aunque suele indentificarse como una evolución del SaaS, es más bien un modelo que ofrece todo lo necesario para soportar el ciclo de vida completo de construcción y puesta en marcha de aplicaciones y servicios web completamente disponibles en internet.

  En ningún momento controlamos las maquinas virtuales, ni atacamos al sistema de las mismas, sino que utilizamos las API de cada nube y sus lenguajes de programación para desarrollar software.

- Infraestructura como servicio (IaaS):

  Modelo de distribución de infraestructura de computación como un servicio, normalmente a través de una plataforma de virtualización (asignación de máquinas virtuales bajo demanda). Incluye servidores, equipamiento de red, almacenamiento,… Estos servicios son proporcionados a través de la web. Múltiples clientes coexisten en la misma infraestructura y se paga por lo que se utiliza.

  Cuando la infraestructura ofrecida es sólo almacenamiento podemos definirlo como "Cloud Storage" o almacenamiento de datos como servicio (Data Storage as a Service). Podemos encontrar desde servicios de almacenamiento de archivos no estructurados hasta bases de datos ofrecidas en la nube. En estos casos, características tales como alta disponibilidad, fiabilidad, replicación y consistencia de datos son muy importantes.

Cada una de estas capas, puede utilizar los servicios de las capas subyacentes:

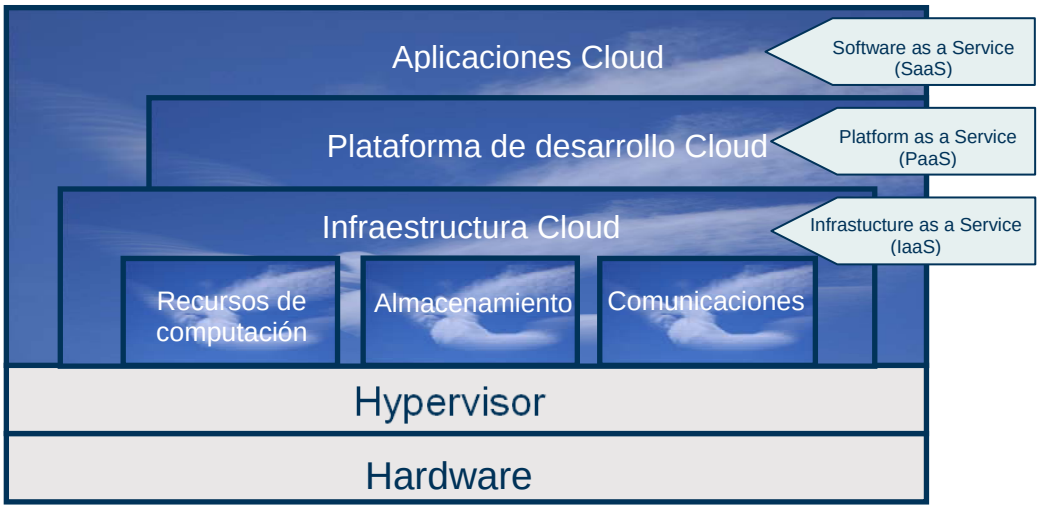

Figura 5 Relación entre capas en Cloud Computing

1.3.3 Clouds públicos, privados e híbridos

- Cloud público:

  Un cloud público es un proveedor de Cloud Computing que ofrece su infraestructura (PaaS o IaaS) al público en general. Proporcionan APIs para gestionar los recursos por parte del cliente.

  Los recursos son dinámicamente aprovisionados en modo autoservicio a través de internet, cogiéndolos del proveedor externo que los factura en función de su uso.

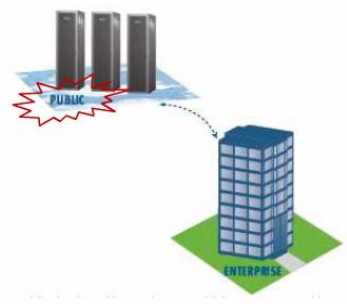

Figura 6 Cloud público

- Cloud privado:

  Tiene características similares al cloud público pero en una red privada. Sirve a una organización con su propia infraestructura y dota de servicios a dicha organización o los clientes de ésta. Las empresas obtienen los beneficios del IaaS pero sin algunas de sus desventajas tales como la privacidad de los datos o la velocidad de acceso a la infraestructura.

Figura 7 Cloud privado

- Cloud híbrido:

  Se denomina cloud híbrido a los servicios de cloud computing que se ofrecen, tanto en modo de pago por uso a clientes (cloud público) como en modo privado sólo para la empresa (cloud privado).

  También puede definirse como un cloud privado que puede nutrirse de recursos de otros clouds, en este caso públicos. Una utilidad de este tipo de organización es la posibilidad de aumentar la capacidad de una infraestructura privada cuando ésta se vea saturada, obteniendo infraestructura de un proveedor de cloud público, para satisfacer demandas pico en momentos puntuales.

  Como se puede observar, es una mezcla entre los dos tipos de nubes anteriores.

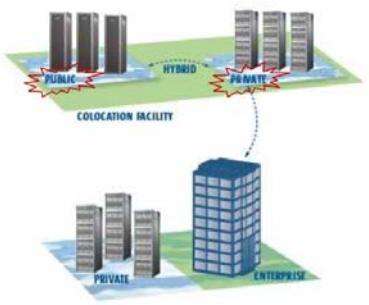

Figura 8 Cloud híbrido

1.3.4 Diferencia entre Grid Computing y Cloud Computing

Una fuente habitual de confusión alrededor del concepto de Cloud Computing es su relación con el Grid Computing. Esta diferencia no está del todo clara quizá porque los clouds y los grids tienen aspectos comunes. Seguidamente, se muestra una tabla donde se encuentran las principales diferencias entre estos dos tipos de computación. La mayor parte de esta tabla está extraída del CloudScape celebrado en Bruselas entre el 14 y el 15 de enero de 2009 ("A workshop to explore the Cloud Computing landscape and its impact on enterprise it") organizado por el OpenGridForum:

Características	Grid	Cloud
Compartición de recursos	Colaboración	Los recursos asignados no son compartidos
Heterogeneidad de recursos	Agregación de recursos heterogéneos	Agregación de recursos heterogéneos
Virtualización	Virtualización de datos y recursos de computación	Virtualización de plataformas hardware y software
Seguridad	Seguridad a través de credenciales	Seguridad a través de aislamiento
Servicios de alto nivel	Bastantes servicios de alto nivel	No hay servicios de alto nivel definidos todavía
Arquitectura	Orientada a servicio	Elegida por el usuario
Dependencias software	Software dependiente del dominio de la aplicación	Software independiente del dominio de la aplicación

Conciencia de plataforma	El software debe estar preparado para su ejecución en un Grid	El software trabaja sobre un entorno personalizado
Flujo de trabajo del software	Las aplicaciones requieren un flujo de trabajo predefinido	Un flujo de trabajo no es esencial para la mayoría de las aplicaciones
Escalabilidad	Nodos escalables	Nodos y hardware escalables
Capacidad	Varía, pero suele ser alta	Bajo demanda
Autogestión	Reconfigurable	Reconfigurable
Centralización	Control descentralizado	Control centralizado
Gestión de fallos	Limitada (a veces las tareas son reiniciadas)	Gran soporte para recuperación de fallos y replicación de contenidos. Las máquinas virtuales pueden ser migradas fácilmente entre nodos
Facilidad de uso	Difícil de gestionar	Cercano al usuario
Estandarización	Estandarización e interoperabilidad	Falta de estándares para interoperabilidad entre Clouds
Acceso para el usuario	Acceso transparente	Acceso transparente

Modelo de pago	Rígido	Flexible (normalmente pago por uso, tipo utility)
QoS	Soporte limitado	Soporte limitado, centrado en disponibilidad

Figura 9 Diferencias entre Grids y Clouds

1.3.5 Ventajas e inconvenientes

A continuación pasamos a analizar las diferentes ventajas e inconvenientes que se nos presentan al adentrarnos en este paradigma.

1.3.5.1 Ventajas

Entre las cuantiosas ventajas del Cloud Computing se encuentran las siguientes:

- Facilidad de escalado: Sólo se utiliza la capacidad que se necesita, en el momento en que se necesita. Las empresas pueden escalar de manera transparente sus servicios adaptándose a las necesidades de sus clientes rápidamente y bajo demanda.
- Posibilidad de trabajar desde cualquier lugar: Sólo es necesario una conexión a internet.
- Ahorro de costes: El pago por uso reduce los costes de manera notable, sobre todo en la fase inicial de las nuevas empresas debido a la reducción de la inversión inicial. Asimismo, no es necesario invertir en infraestructura.
- Ahorro de espacio: No es necesario tener un lugar donde colocar los servidores, ya que accedemos a ellos a través de internet.
- Reducción de las barreras de entrada a nuevos proyectos, negocios y empresas innovadoras.
- Mayor atención al núcleo del negocio: No es necesario preocuparse del mantenimiento de los servidores, compra de hardware, actualizaciones, etc. Esto permite mantener una mayor dedicación al foco principal del negocio.
- Ayuda a gestionar los picos de demanda.

- Mejor utilización de los recursos disponibles: Gracias a la virtualización los recursos se aprovechan al máximo, lo que repercute en los costes y en la ecología.

1.3.5.2 Inconvenientes

Seguidamente, se muestran una serie de inconvenientes o desventajas de este modelo de computación:

- Privacidad: Uno de los mayores miedos que suscita el Cloud Computing es la pérdida de privacidad de tus datos. Al tener que almacenarlos en servidores de otras compañías junto a datos de otros clientes, surge la desconfianza de que alguien pueda ver tu información y utilizarla con fines poco éticos.
- Incumplimiento de la regulación sobre protección de datos: La mayoría de las empresas que ofrecen servicios de Cloud Computing no cuentan con centros de datos en nuestro país, incluso algunas los tienen situados fuera Europa. Esto es un problema cuando los datos no pueden ser almacenados fuera de nuestras fronteras por causas relacionadas con las leyes vigentes.
- Aumento de la latencia: No supone un obstáculo en muchas aplicaciones, pero hay casos en que esta característica puede perjudicar al correcto funcionamiento del sistema.
- Pérdida de control sobre los datos almacenados: Al no estar alojados en nuestros sistemas, el control que tenemos sobre nuestros datos es menor.
- Confidencialidad en la transmisión de los datos a través de conexiones de internet: Son necesarios nuevos sistemas de encriptación más seguros y eficientes.
- Si falla el servicio o la conexión a internet no podemos trabajar.
- Cuidado con la escalabilidad automática: Podemos ser víctimas de ataques de tipo EDoS (Economic Denial of Sustainability). En un ataque tipo DDoS (Distributed Denial of Service), alguien con malas intenciones ataca una aplicación o una web cualquiera con la intención de colapsarla, lanzando peticiones desde múltiples computadoras de manera concurrente y masiva. Al final, el servicio cae ya que no es capaz de copar todas las peticiones solicitadas. El EDoS es una manera más sutil de tirar el servicio abajo. Su objetivo no es la caida física de los servidores que soportan el servicio, sino la rendición de los propietarios del servicio al no poder hacer frente las facturas de uso de los recursos cloud.
- Problema del cliente cautivo: Una vez te has decantado por un servicio, no es sencillo portar lo que has montado en otro proveedor distinto, ya que cada uno tiene unas características diferentes y ofrece unas APIs distintas.

- ¿Qué ocurriría en el caso en el que el proveedor quebrase?

La mayoría de los inconvenientes mencionados arriba sólo aparecen en el caso de cloud públicos.

2 Estado del arte

Este apartado pretende mostrar los diferentes servicios que se ofrecen en la actualidad, relacionados con cada una de las capas de Cloud Computing. De esta forma, tendremos una visión global sobre este modelo de computación y las posibilidades que ofrece.

Figura 10 Principales servicios de Cloud Computing en la actualidad

2.1 Software como servicio

Como se menciona anteriormente, SaaS es un modelo de distribución de software en el que una aplicación se ofrece como un servicio proporcionado a través de internet. Actualmente, existen varios servicios que se ofrecen según este modelo. En este apartado se muestran algunos de los más destacados.

2.1.1 Salesforce

A día de hoy, Salesforce es una de las empresas más notables que ofrecen servicios de SaaS. Fue fundada en 1999 por Marc Benioff, antiguo ejecutivo de Oracle. Ofrecen un CRM (Customer Relationship Management) o software de gestión de clientes accesible desde internet bajo demanda. Se divide en varias aplicaciones, entre las que se encuentran, ventas, marketing, soporte, relación con socios o partners, análisis, etc. Entre sus características más importantes se encuentran:

- Servicio escalable que proporciona mecanismos de recuperación ante fallos.
- Los datos de los clientes están protegidos con medidas de seguridad física, cifrado de datos, autenticación de usuarios, cortafuegos, …
- Integración con otras tecnologías a través de Web-Services.
- Actualizaciones automáticas.
- No es necesario ningún tipo de instalación de software ni de hardware.
- Pago en función del uso. Las actualizaciones están incluidas en la suscripción.
- Acuerdo con Google para la colaboración con Google Apps, de forma que es capaz integrar Salesforce con Gmail, Google Talk, Google Calendar y otros servicios de Google con el fin de cubrir más necesidades empresariales.

Figura 11 Salesforce.com

Como dato curioso, se puede apuntar, que Salesforce.com estuvo caído durante una hora el 6 de enero de 2009 dejando sin servicio a casi un millón de usuarios.

2.1.2 Google Apps

Google es una de las empresas que más fuerte está apostanto por el Cloud Computing. Google Apps es un servicio que ofrece varias aplicaciones accesibles vía web. Es posible acceder a ellas incluso desde dispositivos móbiles.

Entre las herramientas ofrecidas se encuentran algunas tan populares como Gmail, Calendar, Docs, Sites, Talk, Vídeo, una completa API para extender y mejorar los servicios, panel de control para gestionar tu dominio, ayuda, soporte,...

Figura 12 Google Apps

Se ofrecen diferentes ediciones dependiendo de las necesidades del cliente: Standard Edition (gratuita), Premier Edition (de pago) y Education Edition (gratuita).

Para garantizar la seguridad, Google adquirió en 2007 una empresa denominada Postini que trabaja en temas relacionados con la seguridad web.

2.1.3 Windows Live

Figura 13 Windows Live

Se trata de una plataforma en la que Microsoft agrupa un conjunto de servicios y productos software. La mayoría de estos servicios son aplicaciones web accesibles desde un navegador, siendo almacenados el software y la información del usuario en servidores de la compañía estadounidense.

Los servicios que se ofrecen son, entre otros: Windows Live Hotmail, Windows Live Calendar, Windows Live ID, Windows Live SkyDrive, Windows Live Spaces, Windows Live Photos, Windows Live Contacts, Windows Live Maps, Xbox Live, Microsoft Office Live, …

La firma de Bill Gates planea ofrecer tres modelos de precio diferentes para sus servicios bajo demanda:

- El primer nivel sera gratuito y las herramientas online estarán acompañadas de anuncios.
- Habrá un segundo nivel donde se ofrecerán características extra a cambio de un pequeño coste de subcripción.
- Finalmente, existirá un servicio premium para los usuarios con mayores necesidades.

Bill Gates denominó esta apuesta por el Cloud Computing como una revolución en la estrategia de la firma en los últimos años.

2.1.4 MobileMe

Apple también tiene cabida en el mundo del SaaS. MobileMe es una colección de software y servicios online accesibles a través de subcripción. Originariamente se lanzó como iTools y más tarde pasó a denominarse .Mac. Se integra con Mac, PCs (con peor resultado), iTouch y iPhone y es capaz de sincronizar el mail, el calendario, los contactos y las fotos entre todos estos dispositivos. Utiliza el pushmail del iPhone para sincronizarlo con Microsoft Outlook en el PC o con la agenda, iCal y Mail en un Mac. Al comienzo tuvo bastantes problemas (fallos de servidores, problemas de sincronización, emails perdidos, indisponibilidad del servicio,...), aunque poco a poco se van solucionando. Steve Jobs llegó a admitirlo públicamente.

Figura 14 MobiliMe

Está formado por los siguientes servicios:

- Almacenamiento: El plan individual (99$) incluye 20 GB de almacenamiento y 200 GB de transferencia de datos. El paquete familiar (149$) incluye 40 GB de almacenamiento divididos entre una cuenta individual de 20 GB y cuatro cuentas de correo de 5 GB cada una.
- Mail.
- Libreta de direcciones y calendario.
- Galería de fotos.
- iDisk: Repositorio de almacenamiento online accesible mediante un navegador.
- iWeb Publish: Publicación de sitios web.
- iChat.

2.1.5 SAP Business ByDesign

SAP también ofrece un paquete de software de gestión bajo demanda orientado a las pequeñas y medianas empresas que quieran aprovecharse de los beneficios de las aplicaciones de gestión de SAP sin la necesidad de tener una gran infraestructura. Consiste en un conjunto de componentes software online que se integran con el software tradicional de la compañía. Entre otras cosas se ofrece CRM, gestor de finanzas, gestor de proyectos, aplicaciones para recursos humanos, ...

Figura 15 SAP Business ByDesign

2.2 Plataforma como servicio

En esta sección se pretende dar una visión general sobre los PaaS más conocidos a día de hoy. Es curioso ver a empresas cómo Microsoft cuyo modelo tradicional habían sido las aplicaciones de escritorio.

2.2.1 Google App Engine

Google App Engine (GAE) es una plataforma para construcción y almacenamiento de aplicaciones web en servidores de Google. De momento sólo soporta los lenguajes de programación Python y Java, aunque esperan incluir alguno más en un futuro. A continuación se enumeran algunas de sus características más relevantes:

- Las aplicaciones web construidas en GAE escalan automáticamente según las necesidades de tráfico y almacenamiento.
- Por razones de seguridad no proporciona acceso al sistema operativo subyacente a bajo nivel.
- Su punto fuerte es la capacidad de almacenar información en servidores de Google a través de la tecnología BigTable y Google File System (GFS). Para realizar las consultas a Big Table, los ingenieros de Google han creado un lenguaje de consultas, llamado GQL (Google Query Language).
- No sigue el modelo relacional para el almacenamiento de datos, ya que, por ejemplo, no es posible hacer consultas agregadas ni joins.
- El SDK incluye una aplicación de servidor web que emula todos los servicios de App Engine en tu equipo local. También incluye todas las API y bibliotecas disponibles en App Engine.
- Ofrece un API para integrar tu aplicación con las cuentas de Google.
- Proporciona dominios gratuitos (appspot.com), aunque también se pueden utilizar dominios propios.
- Puedes limitar el acceso a tu aplicación a miembros de tu organización.
- Paquete indisoluble: A diferencia de Amazon Web Services, donde puedes utilizar algunos servicios y dejar de lado los demás, GAE no ofrece esa opción.
- Podemos comenzar a utilizar GAE con cuentas gratuitas o adquirir recursos adicionales pagando una cuota, si necesitamos mayor capacidad.
- Proporciona una consola de administración desde donde se pueden gestionar las aplicaciones.

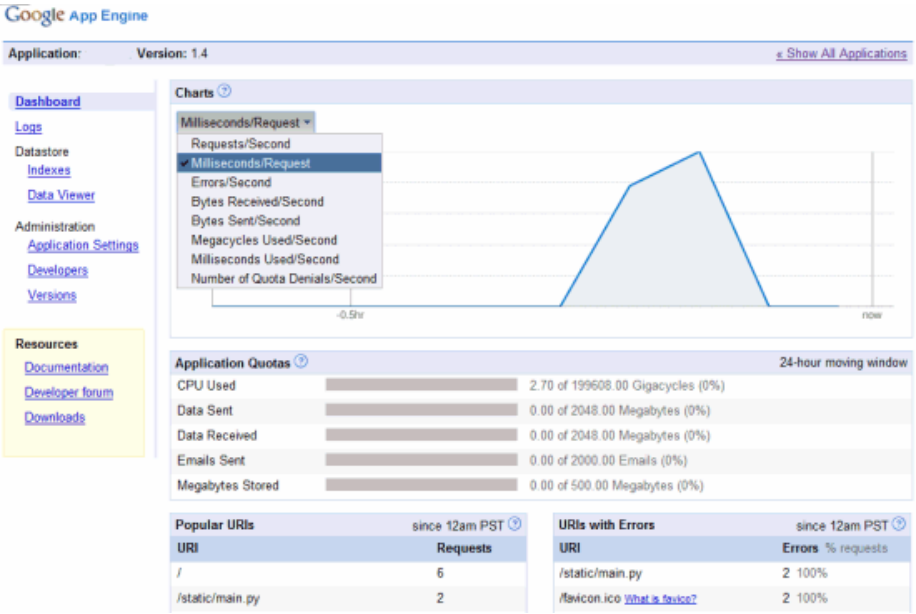

Figura 16 Consola de administración de Google App Engine

2.2.2 Microsoft Azure

Microsoft Azure, también conocido como Azure Service Platform es un PaaS almacenado en los centros de datos de Microsoft que proporciona un sistema operativo (Windows Azure) y un conjunto de servicios que pueden utilizarse individualmente o unidos. Permite a los desarrolladores crear aplicaciones que serán ejecutadas en la nube utilizando sus conocimientos en el entorno de desarrollo Microsoft Visual Studio y el framework Microsoft.NET. Como se puede ver, algunas de las diferencias con Google App Engine son que ofrece un sistema operativo y que se basa en la plataforma .NET en vez de en Python y Java, aunque se espera que en un futuro se soporten más lenguajes de programación y entornos de desarrollo. De momento, están disponibles dos SDKs para interoperabilidad con la plataforma Azure: Java SDK para servicios .NET y Ruby SDK para servicos .NET. Esto permite integrar Java y Ruby con .NET.

Proporciona un entorno con soporte para múltiples protocolos de internet, incluyendo HTTP, REST, SOAP y XML.

Incluye cinco servicios que los desarrolladores pueden utilizar para construir las aplicaciones que se ejucutarán en la nube:

Figura 17 Microsoft Azure

- Windows Azure: Es un sistema operativo orientado a la nube que facilita la gestión de la plataforma Microsoft Azure.
- Live Services: Conjunto de bloques de construcción para gestión de datos de usuario y aplicaciones.
- Microsoft SQL Services: Extiende las posibilidades de Microsoft SQL Server a la nube, convirtiéndola en una base de datos distribuida y relacional basada en la web.
- Microsoft .NET Services: Incluye control de acceso para ayudarte a hacer más seguras tus aplicaciones, un servicio de comunicación entre aplicaciones y servicios,...
- Microsoft SharePoint Services & Dynamics CRM Services: En un futuro, los desarrolladores tendrán acceso a la funcionalidad de SharePoint y de un CRM.

2.2.3 Force.com

Salesforce también tiene su plataforma para desarrollo de aplicaciones en la nube. Force.com incluye base de datos, seguridad, interfaz de usuario y otras herramientas que facilitan la construcción de aplicaciones y sitios web. Estas aplicaciones son construidas utilizando Apex, un lenguaje de programación propietario, similar a Java para la plataforma Force.com y Visualforce (similar a XML) para la construcción de interfaces de usuario en HTML, AJAX o FLEX.

Las aplicaciones construidas sobre este entorno pueden ser distribuidas como SaaS a través de Force.com AppExchange y se integran fácilmente con la aplicación principal de Salesforce.

Figura 18 Force.com

2.2.4 Rollbase

Figura 19 Rollbase

Rollbase es un proveedor de PaaS centrado en la pequeña y mediana empresa. Se pueden crear aplicaciones bajo demanda basadas en AJAX utilizando sencillas herramientas basadas en movimientos de ratón (point & click, drag & drop,…) en un navegador estándar. Se pretende que su utilización sea lo más sencilla posible (no es necesario escribir código), sin perder por ello la posibilidad de crear aplicaciones sofisticadas y de alta calidad.

Es posible la integración con otras aplicaciones a través de los protocolos SOAP y REST. También se ofrece integración con Google Apps.

Entre sus componentes principales se encuentran los siguientes:

- Asistentes para creación de tablas, campos, relaciones, …

- Editor de páginas con posibilidad de introducir scripts y código HTML embebido.
- Motor de búsqueda con control de indexación de campos.
- Integración para aplicaciones construidas en la misma plataforma.
- Gráficos animados 2D y 3D.

2.3 Infraestructura como servicio

En esta sección aparecen algunos de los más importantes proveedores de IaaS. Los que ofrecen almacenamiento como servicio aparecen al final en un subapartado para que quede más claro y ordenado.

2.3.1 Amazon Web Services

Figura 20 Página web de Amazon Web Services

Amazon Web Services (AWS) es una colección de servicios de computación ofrecidos bajo demanda a través de Internet por Amazon.com. Amazon es una compañía estadounidense de comercio electrónico, conocida por la mayoría de los usuarios por la venta de libros a través de Internet. En un momento dado, decidieron obtener beneficios de los grandes centros de datos que tenían repartidos por el mundo y que la mayoría del año estaban infrautilizados y finalmente lanzaron Amazon Web Services en Julio de 2002. La robusta plataforma construida y refinada por Amazon a lo largo de los años, está ahora disponible para todo aquel que disponga de una conexión a Internet.

Estos servicios son accedidos a través de HTTP utilizando los protocolos REST y SOAP y pueden utilizarse unidos o de forma independiente.

Veamos una breve descripción de los principales productos que nos ofrece esta plataforma:

- Servicios de infraestructura:
  - Amazon Elastic Compute Cloud (Amazon EC2):
    - Servicio que permite adquirir máquinas virtuales en pocos minutos y escalar tu capacidad de computación fácilmente, tanto hacia arriba o hacia abajo, bajo demanda.
    - Control total sobre el entorno: Puedes ejecutar las aplicaciones y el software que quieras.
    - Varios tipos de servidores a elegir, con diferentes características. También es posible elegir imágenes predefinidas con un SO y un software preconfigurado.
    - Direcciones IP elásticas: Son direcciones IP estáticas que se asocian a tu cuenta con la posibilidad de controlarlas hasta que decides liberarlas.
    - Balanceo de carga: Elastic Load Balancing, distribuye automáticamente el tráfico de las aplicaciones entre múltiples instancias de EC2.
    - Posibilidad de definir condiciones para escalado automático de forma que sea posible responder satisfactoriamente a picos en la demanda.
    - Pago por uso: Es posible consultar las tarifas en su página web.
  - Amazon SimpleDB:
    - Servicio web para almacenamiento, procesamiento y consulta de datos estructurados. Se explica más detalladamente en el apartado 2.3.5.
  - Amazon Simple Storage Service (Amazon S3):
    - Servicio de almacenamiento bajo demanda. Se explica más detalladamente en el apartado 2.3.5.
  - Amazon CloudFront:
    - Servicio de Content Delivery (distribución de contenidos) capaz de integrarse con los demás servicios de AWS.
    - Actualmente está en estado de beta.

- Amazon Simple Queue Service (Amazon SQS):
  - Infraestructura de colas para almacenamiento de mensajes muy útil para enviar datos y comunicarse entre distintos componentes distribuidos de nuestras aplicaciones.
  - Comunicación entre los diferentes servicios ofrecidos por Amazon de forma que es posible coordinar de una manera eficiente el flujo de trabajo.
  - Mecanismos de autentificación que aseguran que los mensajes almacenados en las colas están protegidos de accesos no autorizados.
- Amazon Elastic MapReduce:
  - Facilita el procesamiento de grandes cantidades de datos.
  - Utiliza un framework Hadoop ejecutándose sobre infraestructura EC2 y S3.
  - Actualmente está en estado de beta.

- Cloud virtual privado:
  - Amazon Virtual Private Cloud:
    - Nuevo servicio, lanzado el 26 de agosto de 2009 y que todavía se encuentra en estado de beta, que permite conectar de forma segura la infraestructura de una compañía con un conjunto de recursos AWS aislados, a través de una conexión VPN (Virtual Private Network o Red Virtual Privada). No es ni más ni menos que la posibilidad de poner una "valla" a las instancias de EC2 utilizadas.
- Pago y facturación:
  - Amazon Flexible Payments Service (Amazon FPS)
  - Amazon DevPay
- …

Actualmente, AWS puede considerarse como el estándar de facto en el mundo del Cloud Computing.

2.3.2 GoGrid

GoGrid es un proveedor de infraestructura cloud, que proporciona máquinas virtuales basadas en Windows y Linux y que pertenece a ServePath, una compañía de hosting estadounidense. Estas máquinas virtuales tienen gran variedad de software preinstalado tal como Apache, PHP, Microsoft SQL Server y MySQL. Además, es accesible a través de un API REST y soporta llamadas desde diferentes lenguajes: Java, PHP, Pitón y Ruby.

Es uno de los mayores competidores de Amazon Web Services, ya que su oferta es de bastante calidad y entre sus características más notables se encuentran las siguientes:

- Soporta entornos de aplicación como Ruby on Rails.
- Cada cuenta ofrece un balanceador de carga hardware F5 gratuitamente.
- Proporciona IPs estáticas y cada cuenta viene con una VLAN pública y privada.
- Interfaz gráfica de usuario desde donde podemos gestionar toda nuestra infraestructura de forma sencilla.
- Podemos elegir entre un amplio rango de imágenes de servidores preconfigurados.
- GoGrid Cloud Storage: Servicio escalable de backup para Windows y Linux a nivel de archivo. Los 10 primeros GB son gratuitos.
- Pago por uso.

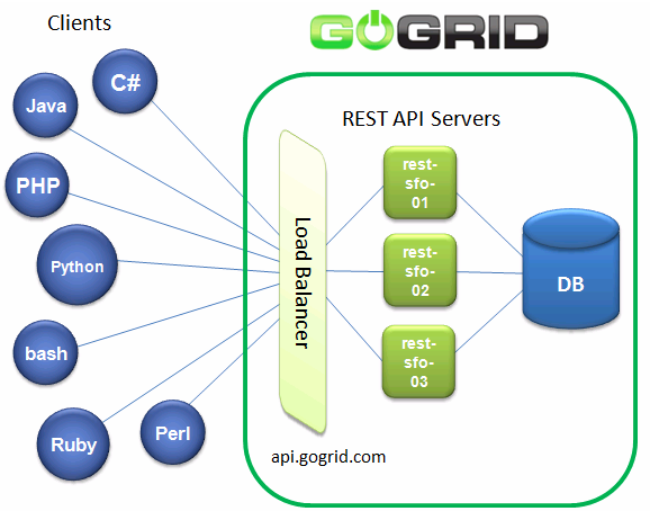

Figura 21 GoGrid

2.3.3 Fexiscale

Flexiscale es la solución de Cloud Computing ofertada por Xcalibre, una empresa dedicada a temas de hosting del Reino Unido.

Figura 22 Flexiscale

Los usuarios de este servicio tienen la posibilidad de crear, arrancar y detener servidores según sus necesidades. Todo ello a través de un API o de un panel de control.

Sus características principales son las siguientes:

- Servidores con soporte para Windows y Linux.
- Recuperación automática antes fallos hardware en menos de 15 minutos respaldada por SLA.
- Ofrece direcciones IP estáticas y una VLAN dedicada para cada cliente.
- Nodos redundantes mediante redes de almacenamiento SAN/NAS.

2.3.4 Mosso

Mosso es una plataforma de IaaS propietaria de Rackspace, una compañía tradicional de web hosting. Al contrario que la mayoría de sus competidores, no ofrece acceso a nivel de root a sus servidores. En vez de eso, se ofrecen servidores preconfigurados con un rango de opciones software y son gestionados de forma similar a un entorno de hosting.

Factura siguiendo un modelo de utility computing y ofrece los siguientes productos:

- Cloud Sites: Hospeda tus sitios web con la posibilidad de escalar a cientos de servidores para ser capaz de responder a las demandas de tráfico existentes. Ofrece base de datos e e-mail.
- Cloud Servers: Instancias de servidores bajo demanda.
- Cloud Files: Ilimitado almacenamiento online.

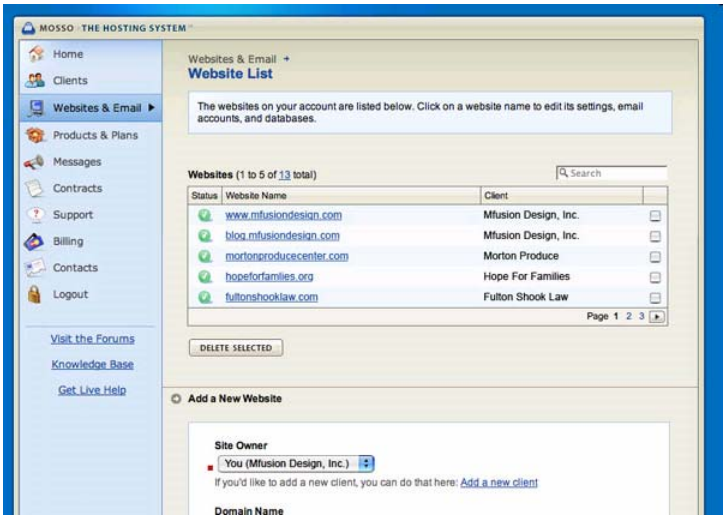

Figura 23 Panel de control de Mosso

2.3.5 Cloud Storage

En este apartado, se muestran algunos de los servicios actuales de *cloud storage* o almacenamiento como servicio. Se encuentran separados de los anteriores, ya que a pesar de ofrecer infraestructura, el concepto no es exactamente el mismo. Mientras los servicios comentados arriba, lo que ofrecen es capacidad de computación a través de la obtención de máquinas virtuales bajo demanda, los que se muestran en esta sección ponen a disposición de sus clientes capacidad de almacenamiento según sus necesidades.

2.3.5.1 Amazon Simple Storage Service (Amazon S3)

Como ya se ha explicado un poco más arriba Amazon S3 es un servicio para almacenamiento y recuperación de datos bajo demanda. Y, ¿qué es lo que ofrece?

- Los datos almacenados pueden ser de cualquier tipo y pueden ser accedidos desde cualquier lugar a través de Internet.
- Se pueden almacenar un número de objetos ilimitado con la restricción de que cada objeto almacenado no puede superar los 5GB.
- Los objetos se almacenan en "buckets", concepto similar al de carpetas en tu sistema operativo.
- Puedes seleccionar la localización donde serán almacenados tus datos. Para ello existen dos opciones, Europa y Estados Unidos.

- Utiliza mecanismos de autenticación para asegurar que los datos están protegidos de forma segura ante accesos no autorizados.
- Se pueden especificar restricciones de acceso para cada objeto en particular.
- Interfaz basada en los estándares REST y SOAP.
- Posibilidad de descargar tus objetos a través del protocolo BitTorrent.

2.3.5.2 Nirvanix

Figura 24 Nirvanix

Nirvanix es una compañía que ofrece servicios de almacenamiento en la nube. Nirvanix's Storage Delivery Network (SDN) es una plataforma de almacenamiento online que, además, proporciona capacidades para manipulación de archivos multimedia (imágenes, audio, vídeo,…).

El acceso a esta plataforma se realiza vía estándares tales como HTTP, REST, SOAP y XML, siguiendo un modelo bajo demanda. Además, proporciona replicación automática de archivos en múltiples localizaciones alrededor del mundo.

Últimamente, esta compañía se ha posicionado como uno de los mayores rivales del servicio S3 de Amazon. Aseguran ser un 200% más rápidos y fáciles de utilizar, aunque algo más caros.

Han desarrollado un sistema NAS orientado a la nube que permite a los usuarios conectar sistemas a los nodos de almacenamiento vía NFS, CIFS y FTP.

2.3.5.3 Amazon SimpleDB

Como se menciona en el apartado 2.3.1, Amazon SimpleDB es un servicio web para almacenamiento, procesamiento y consulta de conjuntos de datos estructurados. Actualmente se encuentra en estado de beta.

Entre sus características principales, destacamos las siguientes:

- No es una base de datos relacional en el sentido tradicional, ya que los datos se almacenan de forma menos estructurada.
- Actualmente, no están permitidas las consultas de referencias cruzadas.
- Proporciona la mayoría de las funciones de una base de datos relacional.
- Una ventaja de no estar fuertemente restringida a los esquemas es la posibilidad de insertar datos al vuelo y añadir nuevas columnas y claves dinámicamente.

2.4 Gestores de clouds

Los gestores de clouds son herramientas software para la construcción y gestión de clouds públicos y privados basados en entornos virtualizados heterogéneos. Crean una capa de virtualización disbribuída, de forma que extienden los beneficios de las plataformas de virtualización a múltiples recursos. Son capaces de desacoplar las máquinas virtuales de su localización física.

Estos gestores, introducen una nueva capa de virtualización entre el servicio y la infraestructura, de forma que se separa la gestión del servicio de la gestión de los recursos de forma totalmente transparente al servicio, y por tanto, a los usuarios finales.

Entre sus beneficios se encuentran:

- Gestión centralizada
- Balanceo de carga
- Consolidación de servidores
- Escalado dinámico
- Particionado dinámico
- Provisión bajo demanda de MVs

Actualmente, no existen muchas opciones a la hora de elegir entre este tipo de gestores, aunque investigando se pueden encontrar algunos. Entre ellos destacan: Eucalyptus, OpenNebula, Abicloud (Abiquo), Enomaly, Qlayer (adquirida por Sun), AppLogic (3Tera), VMware vSphere, Citrix Cloud Center, RetHat Enterprise Virtualization Manager, Aneka, Proxmox Virtual Environment y Platform VM Orchestrator. Cada uno tiene sus propias características. Algunos son bastante limitados, mientras que otros ofrecen multitud de opciones para la gestión de una infraestructura tipo cloud. A continuación se muestran con más detalle algunos de ellos.

2.4.1 OpenNebula

OpenNebula es un gestor de infraestructura virtual de código abierto (open source) desarrollado por el grupo de Arquitectura de Sistemas Distribuidos de la Universidad Complutense de Madrid, que gestiona almacenamiento, red y tecnologías de virtualización y que permite organizar grupos de máquinas virtuales interconectadas, sobre infraestructura distribuida.

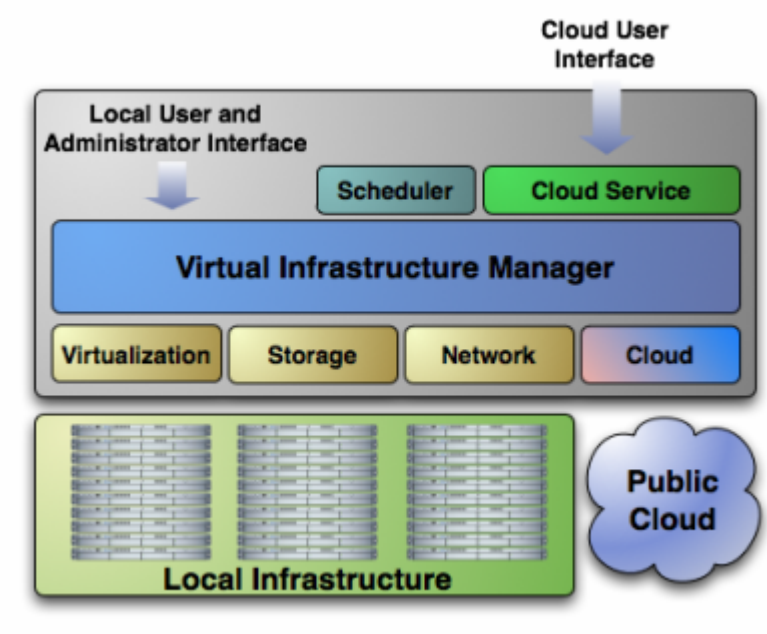

Figura 25 Visión general de OpenNebula

OpenNebula se caracteriza por:

- Transforma una infraestructura distribuida en una infraestructura virtual flexible (cloud privado), capaz de ajustarse dinámicamente a fluctuaciones en la demanda.
- Combina infraestructura local con intraestructura pública ofertada por proveedores de Cloud Computing (conectores para Amazon EC2 y ElasticHosts) consiguiendo entornos altamente escalables (cloud híbrido).

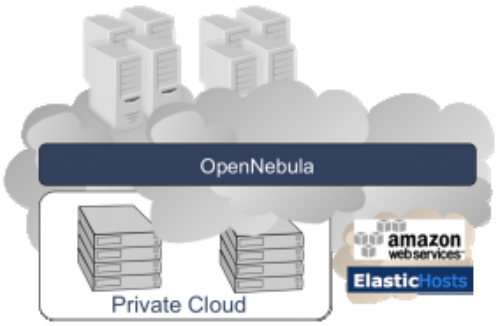

Figura 26 Construcción de nubes híbridas con OpenNebula

- También incluye la posibilidad de construir cloud públicos, ya que proporciona interfaces para externalizar la capacidad de gestión de los recursos virtuales.
- API XML-RPC para administradores y usuarios.
- Soporta las plataformas de virtualización Xen, KVM, VMWare y próximamente Virtual Box. También proporciona un conector a libvirt.
- Aporta mecanismos para transferencia y clonación de imágenes de máquinas virtuales.
- Posibilidad de definir redes virtuales aisladas para interconectar máquinas virtuales.
- Tolerancia a fallos.
- Arquitectura abierta, flexible y extensible.
- Numerosos conectores y acuerdos con empresas y organizaciones que proporcionan características adicionales. Entre ellos podemos citar el planificador Haizea desarrollado en la universidad de Chicago.
- Se enmarca dentro del "Reservoir Project", proyecto europeo sobre infraestructuras virtualizadas y Cloud Computing.

2.4.2 Eucaliyptus

EUCALYPTUS (Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems) es un sistema open source para implementación de clouds privados e híbridos utilizando tu propia infraestructura que nació en la universidad de California. Proporciona un interfaz compatible con el de Amazon EC2, aunque su arquitectura está diseñada para ser modificada y extendida de forma que pueden ser soportadas múltiples interfaces.

Figura 27 Eucalyptus

Está implementado utilizando herramientas Linux y tecnologías de servicios web. Su arquitectura está diseñada para tratar a los nodos como recursos y cada tarea de procesamiento se divide en componentes. El controlador gestiona los recursos.

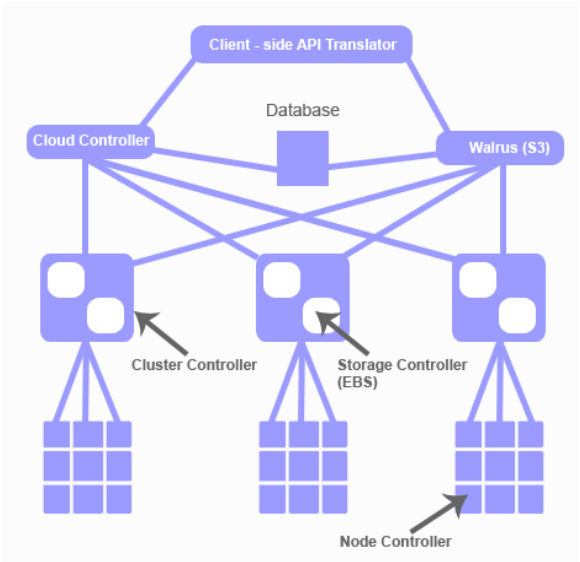

Figura 28 Arquitectura de Eucalyptus

Actualmente soporta los hypervisores Xen y KVM y garantiza comunicación interna segura utilizando SOAP.

2.4.3 AbiCloud

Abicloud es un software de infraestructura opensource para la creación y gestión integral de clouds públicos y privados basados en entornos virtualizados heterogéneos que pertenece a la empresa española Abiquo.

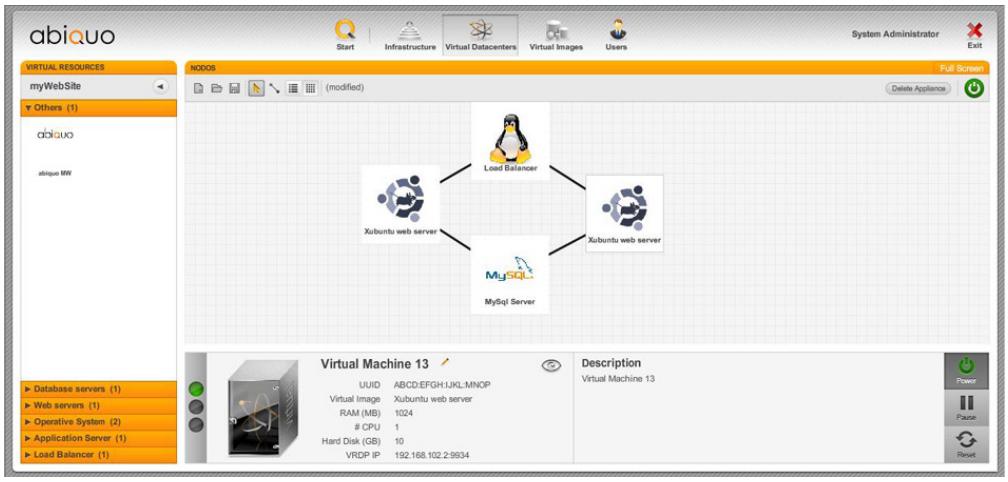

Figura 29 Interfaz gráfica de Abicloud

Esta herramienta ofrece a los usuarios principalmente, capacidad de escalado, gestión y provisión automática e inmediata de servidores, almacenamiento, redes, dispositivos de red virtuales, así como de aplicaciones. Además, se caracteriza por ofrecer una rica interfaz gráfica de usuario que facilita su gestión.

A día de hoy soporta los hypervisores Xen, KVM, VMware y VirtualBox, aunque se espera que dentro de poco sea capaz de soportar Hyper-V también. Asismismo, la plataforma es capaz de conectarse a múltiples repositorios remotos de imágenes almacenadas en formato OVF (Open Virtualization Format). La figura siguiente muestra de forma resumida el diagrama con los principales componentes de la arquitectura:

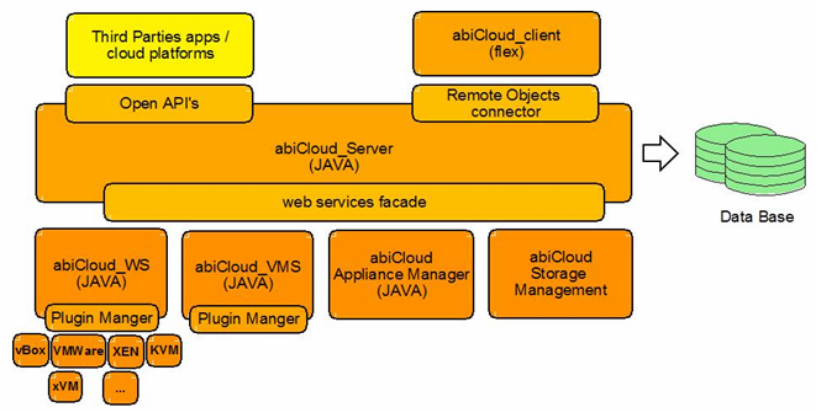

Figura 30 Diagrama arquitectónico de Abicloud

2.4.4 Enomaly's Elastic Computing Platform

Enomaly's Elastic Computing Platform (ECP) es una herramienta para la creación de servicios de infraestructura bajo demanda tipo Cloud Computing. Incluye un motor automatizado para configurar, administrar y desplegar grupos de máquinas virtuales. Para ello, proporcionan un API y una interfaz gráfica de usuario que facilita las operaciones con el entorno.

Figura 31 Interfaz gráfica de Enomaly's Elastic Computing Platform

Gestión de la red virtual, gestión de repositorios de máquinas virtuales, creación de imágenes, seguridad y permisos, motor de reglas de provisión y capacidad de creación de nubes híbridas son algunas de sus características.

2.4.5 3Tera AppLogic

Figura 32 3Tera

AppLogic es un sistema que convierte un conjunto de servidores en una infraestructura virtualizada siguiendo un modelo grid/cloud. No sólo se pueden definir máquinas virtuales, sino también infraestructuras más complejas cómo cortafuegos, VPNs, balanceadores de carga y almacenamiento, etc. Y todo ello desde un navegador.

AppLogic empaqueta todo el código, los datos y la infraestructura requerida para ejecutar una aplicación web escalable, en una única entidad lógica que puede ser arrancada, parada, gestionada, copiada e incluso exportada a otra infraestructura sin modificaciones.

2.5 ¿Quién es el líder?

Echando un vistazo a los apartados anteriores, nos podemos hacer una idea de la amplia oferta existente en el espacio del Cloud Computing, y eso teniendo en cuenta que sólo aparecen unos pocos dentro del amplio ecosistema actual. Viendo este panorama, surge la cuestión de quienes serán los que lleven la "voz cantante" en un mercado con tanta competencia.

En el CloudCamp ("Inside the Cloud", Cloud Computing Community Survey) celebrado el 19 de noviembre de 2008 realizaron un estudio donde se puede apreciar quien es el lider de la industria en estos momentos. La gráfica siguiente muestra los resultados de este estudio:

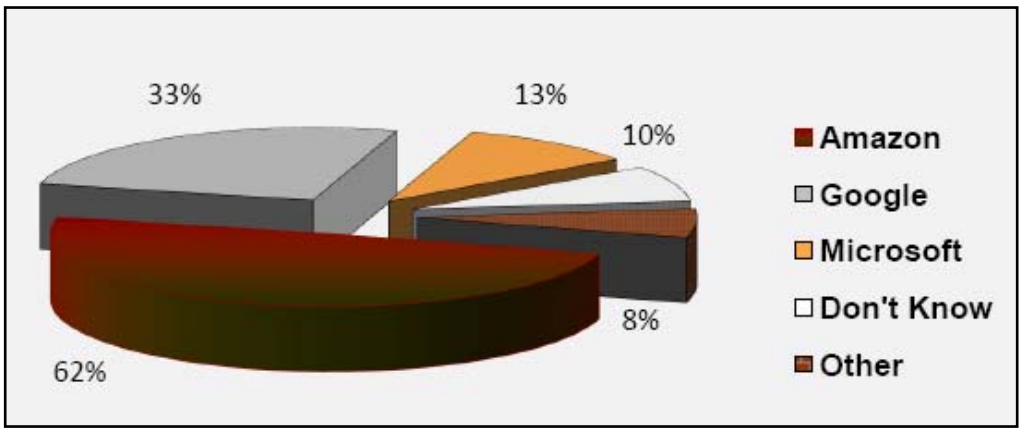

Figura 33 ¿Quién es el líder?

Como se puede observar, Amazon encabeza los resultados, llegando casi a doblar a Google que aparece en segundo lugar. En tercer lugar se encuentra Microsoft, aunque bastante alejado de los dos primeros, y los porcentajes restantes los forman empresas tales como Sun, IBM y AT&T.

En este estudio también se concluye que los proveedores de infraestructura lideran la innovación en el campo del Cloud Computing por delante de los proveedores de plataforma y aplicaciones.

2.6 Arquitectura general del espacio cloud. Modelo de referencia

En este apartado se pretende dar una visión general de la arquitectura del Cloud Computing. Para ello, se ha elaborado un diagrama donde se han intentado plasmar todos los componentes que aparecen en los distintos sistemas cloud. Además, se ha dividido en capas, de forma que sea posible asociar cada uno de estos componentes con el tipo de Cloud Computing al que pertenecen.

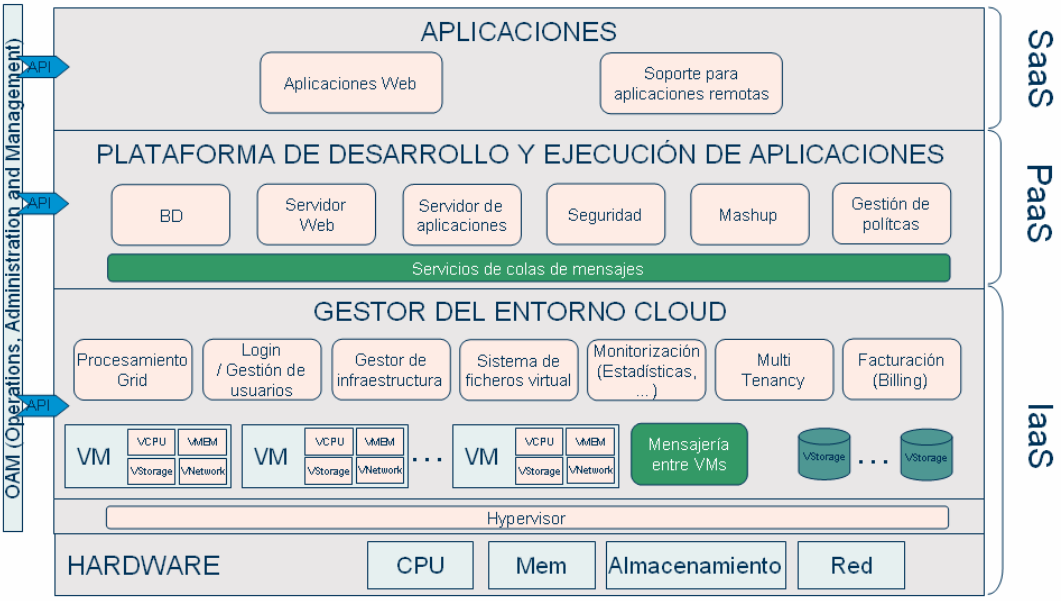

Figura 34 Arquitectura general del espacio Cloud

Como vemos, en la zona inferior del diagrama se encuentra la parte de hardware. Desplazándonos hacia arriba, nos encontramos con el hypervisor, cuya función es la creación de una capa de virtualización disbribuída. Esta capa virtual (máquinas virtuales, redes virtuales, etc) necesita ser gestionada mediante una herramienta de gestión de entornos virtuales tipo cloud. Podría ser alguna de los vistas en el apartado 2.4. El componente principal de esta herramienta es el gestor de infraestructura, que a su vez, está formado por varios componentes que se muestran en la figura 35.

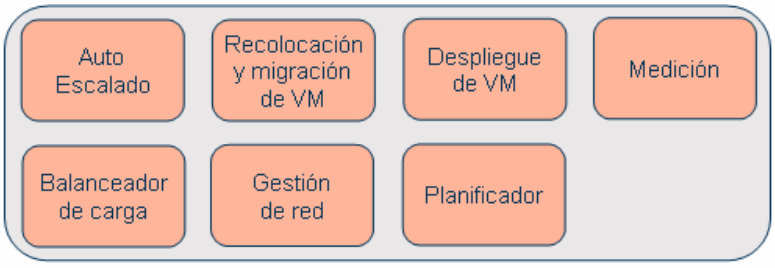

Figura 35 Gestor de infraestructura

Siguiendo hacia arriba, tenemos un servicio de colas de mensajes, que es el encargado de facilitar la comunicación entre los distintos servicios ofrecidos. La siguiente capa, reúne todos "ingredientes" necesarios para soportar el ciclo de vida completo de una aplicación en la nube. Finalmente, encontramos la capa de aplicaciones.

En la parte izquierda del diagrama aparece un módulo en color azul. Su función es la de proporcionar una serie de herramientas (interfaces gráficas, APIs, etc) al usuario de la arquitectura cloud para que sea capaz de administrarla y operar con ella fácilmente.

3 Estudio comparativo entre diferentes plataformas

En la actualidad, existen una gran cantidad de herramientas y servicios relacionados con el Cloud Computing, según hemos podido comprobar más arriba. Hemos elegido algunas de las más significativas para hacer una comparación que nos ayudará en la elección de una de ellas para la construcción de nuestra arquitectura.

En primer lugar, mostramos una comparativa general de aspectos comunes a todas ellas. Seguidamente, las dividimos en tres grupos. En primer lugar, comparamos las soluciones de IaaS y PaaS. A pesar de ser dos conceptos diferentes, es posible hacer una comparación entre ellas en la misma tabla. En segundo lugar aparece la comparativa entre servicios de almacenamiento de datos en la nube, y para terminar, se contrastan dos gestores de infraestructura cloud.

3.1 Comparación general

Amazon EC2, GoGrid, GoogleAppEngine, Amazon S3, Nirvanix SDN, OpenNebula y AbiCloud son las seis opciones elegidas para realizar esta comparativa. Se puede ver, que existe al menos un representante de cada uno de los tipos de Cloud Computing existentes en la actualidad. Además, cada uno de estos representantes está entre los más destacados y populares dentro de su grupo, lo que ha sido determinante a la hora de decantarnos por ellos.

	Amazon EC2	**GoGrid**	**Google App Engine**	**Amazon S3**	**Nirvanix SDN**	**Open Nebula**	**AbiCloud**
Tipo	IaaS	IaaS	PaaS	IaaS (Cloud Storage)	IaaS (Cloud Storage)	Gestor	Gestor
Breve descripción	Servicio web que permite la adquisición de máquinas virtuales para proporcionar una arquitectura escalable bajo demanda	Proveedor de servicios de infraestructura, que proporciona máquinas virtuales bajo demanda gestionadas a través de un panel de control	Plataforma para construcción y almacenamiento de aplicaciones web en servidores de Google	Servicio web para almacenamiento y recuperación de datos	Plataforma de almacenamiento online	Motor de ifraestrctura virtual que posibilita la gestión dinámica de máquinas virtuales sobre un cluster de recursos físicos	Software para la creación y gestión integral de clouds públicos y privados basados en entornos virtualizados heterogéneos

Localización de servidores	EEUU y Europa	EEUU	Google no informa sobre donde tiene almacenados tus datos. No revela si tus datos residen en EEUU o si se encuentran dispersados geográficamente	EEUU y Europa	EEUU, Asia y Europa	No aplicable	No aplicable
Open Source	No	API liberada bajo licencia Creative Common	No	No	No	Si	Si
Soporte	400$/mes (24/7)	Gratuito (24/7)	En este momento no tienen disponible ni email, ni teléfono para realizar preguntas. Únicamente disponen de Google App Engine FAQs	400$/mes (24/7)	Soporte standadr (e-mail, foro y chat) y soporte enterprise (24/7)	Lista de correo (Support mailing list)	Ofrece diferentes modelos de soporte a sus productos en función de las necesidades de cada cliente
Modelo de facturación	Instancias bajo demanda (pago por uso) e instancias reservadas	Pago por uso y prepago	Cuotas libres y cuotas de pago (pago por uso)	Pago por uso	Pago por uso	Gratuito	De momento se puede descargar gratuitamente, ya que la versión actual no es aún definitiva

Figura 36 Comparativa EC2, GoGrid, GAE, S3, SDN, OpenNebula y Abicloud

3.2 IaaS & PaaS

La siguiente tabla muestra la comparativa entre los servicios de plataforma e infraestructura ofrecidos por Amazon, GoGrid y Google.

	Amazon EC2	**GoGrid**	**Google App Engine**
Virtualización	Hypervisor Xen (paravirtualización)	Hypervisor Xen (virtualización completa)	Contenedor de aplicaciones

Escalabilidad	Arquitectura elástica y escalable. El cliente puede preparar nuevas máquinas o éstas pueden ser proporcionadas por alguna herramienta tal como RightScale	Su interfaz permite añadir y borrar servidores instantaneamente	Escalabilidad automática (+,-) , tanto en potencia de computación como en almacenamiento
Características de almacenamiento	AWS ofrece dos servicios adicionales para almacenamiento, Amazon S3 para datos no estructurados y Amazon SimpleDB para BDs	GoGrid Cloud Storage (servicio escalable de backup a nivel de archivo para servidores ejecutado sobre GoGrid) Los primeros 10GB son gratuitos	Datastore basado en BigTable (BD no relacional) y sistema de ficheros basado en Google File System (GFS)
Rendimiento	Depende del tipo de instancia. Usualmente, estas instancias tardarán unos minutos en levantarse	Depende del tipo de instancia. Usualmente, estas instancias tardarán unos minutos en levantarse	Infraestructura de alto rendimiento de Google
Redundancia	Proporciona la posibilidad de colocar instancias en múltiples localizaciones	Puedes levantar otra instancia de una imagen y añadirla al balanceador de carga	BigTable y Google File System (GFS)
Balanceo de carga	72$/mes	Balanceador de carga hardware F5 gratuito	Gratuito
Tolerancia a fallos	Direcciones IP elásticas y zonas de disponibilidad (Availability zones)	Múltiples instancias y balanceador de carga	Depende de la infraestructura de Google
Sistemas operativos y software instalados	Red Hat Enterprise Linux Windows Server 2003 Oracle Enterprise Linux OpenSolaris openSUSE Linux Ubuntu Linux Fedora Gentoo Linux Debian Oracle 11g Microsoft SQL Server Standard 2005 MySQL Enterprise Microsoft SQL Server Express Hadoop Condor Open MPI Apache HTTP IIS/Asp.Net Java Application Server JBoss Enterprise Application Platform Ruby on Rails Wowza Media Server Pro Windows Media Server	Windows Server 2003 Windows Server 2008 RedHat Linux CentOS SQL Server 2005 SQL Server 2008 Postgree SQL 8.1 MySQL 5.0 Apache 2.2 PHP 5.1 PHP 5.2 ASP.NET Fast CGI IIS 7	Invisible al programador

Imágenes personalizadas	Amazon EC2 ofrece la posibilidad de construir y personalizar imágenes (Amazon Machine Images - AMIs-)	No soporta imágenes personalizadas	Característica no aplicable. GAE es un PaaS, no un IaaS
Direcciones IP de servidores	Direcciones IP elásticas (direcciones IP estáticas diseñadas para esta plataforma dinámica). Una dirección IP elástica se asocia con tu cuenta (no con una instancia particular) y tienes la posicilidad de controlarla hasta que la liberas	GoGrid proporciona direcciones IP estáticas y cada cuenta viene con una VLAN pública y privada. Por defecto, la VLAN pública tiene 16 direcciones IP y la privada 256	Invisible al programador
Características de los servidores disponibles	Small Instance: 1.7 GB RAM, 1 EC2 Compute Unit (1 virtual core con 1 EC2 Compute Unit*), 160 GB Disk, 32-bit Large Instance: 7.5 GB RAM, 4 EC2 Compute Units (2 virtual cores con 2 EC2 Compute Units cada uno), 850 GB Disk, 64-bit Extra Large Instance: 15 GB RAM, 8 EC2 Compute Units (4 virtual cores con 2 EC2 Compute Units cada uno), 1690 GB Disk, 64-bit High-CPU Medium Instance: 1.7 GB RAM, 5 EC2 Compute Units (2 virtual cores con 2.5 EC2 Compute Units cada uno), 350 GB Disk, 32-bit High-CPU Extra Large Instance: 7 GB RAM, 20 EC2 Compute Units (8 virtual cores con 2.5 EC2 Compute Units cada uno), 1690 GB Disk, 64-bit *EC2 Compute Unit (ECU) – Equivalente a la capacidad de una CPU de 1.0-1.2 GHz 2007 Opteron o un procesador Xeon 2007	1 x Xeon, 5 GB RAM, 30 GB Disk 1 x Xeon, 1 GB RAM, 60 GB Disk 1 x Xeon, 2 GB RAM, 120 GB Disk 3 x Xeon, 4 GB RAM, 240 GB Disk 6 x Xeon, 8 GB RAM, 480 GB Disk	Característica no aplicable. GAE es un PaaS, no un IaaS
Seguridad	Firewall configurable que controla el acceso a la red y entre grupos de instancias	Tienes que instalar un firewall y mantenerlo tú mismo	Google Secure Data Connector
SLA	Si (99.95%)	Si (10,000%)	No
Interfaz de usuario, administración y monitorización	Linea de comandos	GUI basada en AJAX o API	Consola de administración web
APIs y protocolos de acceso	API basada en SOAP	API basada en REST. Las llamadas de los métodos se realizan enviando peticiones HTTP GET o	GAE soporta los métodos HTTP GET, POST, PUT, DELETE, OPTIONS y HEAD pero no tiene una interfaz

		POST al servidor de REST	SOAP ni REST. El entorno de ejecución de Python tiene APIs para los siguientes servicios: Datastore, Memcache, URL Fetch, Mail, Images, Google Accounts
Lenguajes y frameworks soportados	C++, C#, Java, Perl, Python, Ruby	Java, PHP, Python, Ruby, Perl, C#	Actualmente, GAE permite escribir tus aplicaciones solo con Python y Java. Adicionalmente, las plantillas de tus sitios web pueden incluir Javascript junto con tu HTML Permite la mayoría de los frameworks de Python. Django está incluido en el SDK
SSH	Si	Si	No
FTP	Si	Si	No
Hardware configurable	Si	Si	No
Acceso de root	Si	Si	No
Direcciones IP privadas en VLANs separadas	No	Si	Invisible al programador
Protección DoS	Para mitigar potenciales ataques DDoS	Gratuito	Max Daily Budget
Otras características	No es demasiado fácil de usar AWS Toolkit para Eclipse Otros servicios compatibles (Amazon SimpleDB, Amazon CloudFront, Amazon SQS, AWS Premium Support, Amazon DevPay, …)	Fácil de utilizar Cloud Connect (Permite construir infraestructura híbrida)	Fácil de utilizar

Figura 37 Comparativa Amazon EC2, GoGrid y GAE

3.3 Cloud Storage

En este apartado se contrastan las soluciones de almacenamiento en la nube de Amazon y Nirvanix, dos de las más completas en su sector.

	Amazon S3	Nirvanix SDN
Modelo de almacenamiento	Amazon S3 es un servicio de almacenamiento accesible a través de internet. Puede ser utilizado para almacenar y recuperar cualquier cantidad de datos, en cualquier momento y desde cualquier sitio, a través de la web. Su objetivo es maximizar los beneficios del autoescalado y hacerlos disponibles al desarrollador	Nirvanix's Global Storage delivery Network (SDN) proporciona almacenamiento inteligente en la nube basado en políticas. SDN almacena, despliega y procesa peticiones inteligentemente en la mejor localización de la red. Utiliza el sistema de ficheros Internet Media File System (IMFS), un sistema de archivos distribuido que incluye a todos los nodos de almacenamiento distribuidos bajo un mismo espacio de nombres
Escalabilidad	Almacenamiento escalable bajo demanda. Amazon S3 puede escalar en términos de almacenamiento, tasa de peticiones y número de usuarios. Añadir nodos al sistema incrementa su disponibilidad, velocidad , rendimiento, capacidad y robustez	Ilimitada escalabilidad bajo demanda dirigida por hardware y software
Capacidad	Capacidad de almacenamiento virtualmente ilimitada	Puede escalar hasta petabytes
Latencia	Amazon S3 debe ser suficientemente rápido para soportar aplicaciones de alto rendimiento. La latencia por parte del servidor es insignificante en contraste con la relativa a internet. Cualquier cuello de botella puede ser reparado añadiendo nodos al sistema	Su arquitectura de nodos soporta enrutado inteligente de archivos a las localizaciones más cercanas para acelerar el rendimiento. Al mismo tiempo, IMS no es la mejor opción para aplicaciones con excesivas operaciones de entrada salida. Estas aplicaciones requieren tiempos de respuesta por debajo del milisegundo que internet no es capaz de soportar todavía
Redundancia	Los datos almacenados son replicados en múltiples servidores	Permite replicación de datos a través de múltiples nodos geográficamente dispersados
Alta disponibilidad	99,99%	99,9% (1 nodo) – 100% (3 nodos)
Balanceo de carga	No	Balanceo de carga dinámico dentro de un mismo nodo y entre distintos nodos
Tolerancia a fallos	Técnicas no centralizadas para eliminar fallos locales y cuellos de botella al escalar	Replicación basada en políticas que evita a los usuarios ser vulnerables antes fallos locales
Seguridad	Amazon S3 es accesible vía SSL. Las peticiones son autorizadas utilizando listas de control de acceso (ACL) asociadas con cada objeto	Soporte SSL y modelo de autenticación de API
Crear, obtener y borrar archivos	Si	Si
Renombrar, mover y copiar archivos	No	Si
Políticas de limitación de uso y ancho de banda	No	Si

SLA	Si	Si
Interfaz de usuario, administración y monitorización	Interfaz basada en web services	Ofrece múltiples elecciones: API basada en estándares, proxy FTP y CloudNAS (software que mapea Nirvanix como un disco en Windows y Linux). Nirvanix Management Portal (NMP) proporciona información y opciones de administración al servicio
APIs y protocolos de acceso	Interfaces REST y SOAP diseñadas para trabajar con cualquier herramienta de desarrollo a través de internet	API REST y SOAP
Procesamiento multimedia para reescalado y rotación de imágenes, codificación de vídeo, …	No	Si
Compartición de archivos y carpetas	Si	Si
Tamaño máximo de archivo	5 GB	256 GB
Otras características	Puedes utilizar el protocolo BitTorrent para recuperar cualquier objeto público en S3	Nirvanix Migration Tool (herramienta para la transferencia de archivos de S3 a Nivanix manteniendo la estructura)

Figura 38 Comparativa Amazon S3 y Nirvanix SDN

3.4 Gestores de infraestructura virtual

A continuación comparamos dos plataformas open source bastante completas para la construcción de clouds desarrolladas en España. OpenNebula, desarrollada por el grupo de investigación de arquitectura de sistemas distribuidos (DSA- Research) de la Universidad Complutense de Madrid y Abicloud, creada por la joven empresa barcelonesa Abiquo.

	OpenNebula	**AbiCloud**
Hypervisores soportados	Xen, KVM y VMWare (próximamente estará disponible también VirtualBox). También ofrece un conector genérico libvirt	VirtualBox, KVM, Xen y VMware ESXi
Escalabilidad	Provisión bajo demanda de máquinas virtuales. Ha sido testeado con cientos de servidores y máquinas virtuales	Provisión bajo demanda de máquinas virtuales. Ha sido testeado con cientos de máquinas virtuales y menos de 100 servidores

Interfaz de usuario, administración y monitorización	Potente API y CLI (Command Line Interface) para monitorización y control de máquinas virtuales y recursos físicos. Además, ofrece el API de virtualización de libvirt	Rica interfaz web de usuario desarrollada en Adobe Flex y personalizable, al ser opensource
Tipo de gestión	Centralizada	Centralizada
Gestión de imágenes	Mecanismos para migración y clonación de imágenes	Mecanismos para almacenamiento y gestión de imágenes en repositorios. Permite montar y clonar las imágenes directamente desde los repositorios
Balanceo de carga	Si	Si
Red	El módulo de gestión de red ofrece la posibilidad de definir redes privadas virtuales y administrarlas. Asignación de IPs mediante DHCP o asignación fija	El módulo de gestión de red ofrece la posibilidad de definir redes privadas virtuales y administrarlas. Asignación de IPs mediante DHCP o asignación fija
Soporte para cargas de trabajo heterogéneas	Si	Añade los conceptos de aplicación virtual y centro de datos virtual. Las aplicaciones virtuales pueden ejecutarse aisladas en centros de datos virtuales
API y protocolos de acceso	XML-RPC API	API para gestión remota de la plataforma
Tolerancia a fallos	Backend de base de datos persistente para almacenamiento de información sobre hosts y máquinas virtuales	No
Seguridad	Gestión de usuarios por parte del administrador de la infraestructura	Gestión de usuarios, grupos y sus permisos
Interfaces para la construcción de nubes públicas	Subconjunto del API de consulta de Amazon y del API OGF OCCI (Open Cloud Computing Interface)	No
Acceso a proveedores de cloud computing remotos	Amazon EC2 y ElasticHosts	No
Tecnologías para almacenamiento	NFS	iSCSI, NFS, CIFS
Herramientas que extienden la funcionalidad	Haizea: Extiende las capacidades de planificación y definición de políticas de reserva avanzada de recursos Virtual Cluster Tool: Añade soporte para la gestión de clusters virtuales	abiNtense: Herramienta para la construcción de entornos Grid abiData: Solución de almacenamiento en la nube de bajo coste
Otras características	Utilizado en varios proyectos internacionales sobre Cloud Computing y virtualización. Es el gestor de infraestructura virtual en el proyecto RESERVOIR (proyecto europeo sobre infraestructuras virtualizadas y Cloud Computing)	Las nuevas máquinas virtuales creadas con AbiCloud se asignan de forma transparente al usuario a la primera máquina física que sea capaz de ejecutarlas. En este sentido es diferente de OpenNebula, donde es necesario indicar donde serán ejecutadas

Figura 39 Comparativa OpenNebula y Abicloud

4 Elección de las tecnologías y construcción del entorno

En este capítulo vamos a explicar los detalles sobre el entorno cloud construido y la base de datos desplegada sobre él. En el primer apartado, se han tenido en cuenta una serie de aspectos relacionados con las bases de datos y la gestión de éstos (data management) que pueden verse afectados por este tipo de arquitecturas. De esta forma, seremos capaces de darnos cuenta de cómo afectan (positiva o negativamente) este tipo de modelos de computación en aplicaciones de esta clase. En los apartados siguientes aparece el proceso de creación del entorno desde el principio, comenzando por la elección de las tecnologías y finalizando con los pasos seguidos para su construcción.

4.1 Aspectos de data management afectados por arquitecturas tipo cloud

Muchos son los aspectos que pueden verse afectados al ejecutar una base de datos sobre un entorno distribuido tipo Cloud Computing. Estos entornos tienen algunas características que los diferencian de los sistemas tradicionales, y por tanto, al desplegar ciertos tipos de aplicaciones, éstas pueden verse afectadas, tanto para bien como para mal. A continuación se muestran los principales aspectos de data management que pueden verse influidos por este tipo de sistemas. Este listado, será de gran utilidad a la hora de evaluar el comportamiento de la BD sobre el entorno cloud construido.

- Escalabilidad: Puede verse afectada de forma positiva al ser ejecutada en un entorno escalable dinámicamente. Sin embargo, es necesario que la BD tenga la capacidad de añadir nodos en caliente.
- Latencia: Aspecto importante a estudiar en este contexto. Debido a las nuevas capas introducidas (virtualización, gestor de infraestructura virtual,…) puede verse aumentada. Además, en caso de cloud públicos, el ancho de banda de la conexión a internet también será un inconveniente.
- Rendimiento: Temas como escalabilidad, latencia, localización de los datos, balanceo de carga, etc, pueden hacer que este factor se vea beneficiado o perjudicado en este tipo de arquitecturas.
- Alta disponibilidad: En el caso de cloud públicos, es muy importante que el proveedor del servicio ofrezca un SLA con altos porcentajes de disponibilidad.

- Tolerancia a fallos: Es fundamental, que el entorno donde sea ejecutada la BD sea lo más robusto posible. Aunque este tema no sólo depende del entorno cloud, sino también de los propios mecanismos ofrecidos por la BD y de como sean utilizados.
- Seguridad: Aspecto bastante importante. Son de vital importancia soporte SSl, encriptación, aislamiento de los datos, modelos de autenticación, listas de control de acceso (ACLs) y cualquier mecanismo que garantice la seguridad dentro del sistema.
- Localización de los datos: Influirá en aspectos tales como latencia y rendimiento.
- Multy-Tenancy: Al poder ser ejecutadas aplicaciones de distintos usuarios sobre los mismos recursos, es muy importante que los programas y datos de cada uno estén perfectamente aislados y protegidos.
- Balanceo de carga: Tanto la BD, como el hardware, como los propios gestores de la infraestructura virtual deben proporcionar mecanismos para un mejor funcionamiento del sistema en general.
- Transaccionalidad: Depende de la BD, aunque pudiera verse afectada por la arquitectura.
- Replicación: Depende tanto de la BD como del entorno cloud. Sería bueno, que el sistema cloud proporcionase mecanismos de replicación de datos y de máquinas virtuales para proporcionar una mayor robustez.
- Consistecia de datos: Fundamental en sistemas distribuidos tales como los sistemas cloud. Aspecto muy relacionado con las propiedades ACID de la BD tales como la transaccionalidad. Existen varios modelos que permiten garantizarla: consistencia secuencial, consistencia eventual, consistencia delta, consistencia débil,…
- Interfaz de acceso: Facilita la vida al usuario. Es recomendable que sea fácil de usar y ofrezca varias posibilidades.
- Facturación (billing): Importante sólo en el caso de que se quiera cobrar por los servicios construídos.
- Datos cautivos: Este aspecto es muy importante. Sería aconsejable, que existieran unos estándares abiertos para poder interoperar entre distintos proveedores de cloud computing, de forma que los usuarios y sus datos no fuesen cautivos del proveedor al que pertenecen. Actualmente, existen varios organismos que se encargan de este tema.

4.2 Elección de las tecnologías

La primera decisión que hubo que tomar fue la de elegir entre desplegar la BD sobre un cloud público o construir un cloud privado.Al final se decidió construir un cloud privado, debido a que de esta forma tendríamos un mayor control sobre la infraestructura y sobre los datos almacenados en ella. Además, la latencia no se vería tan afectada, uno de los aspectos fundamentales en una base de datos. Una vez se tuvo claro este aspecto, fue necesario elegir entre distintos componentes necesarios para la construcción del entorno. Una de las cosas que estaban claras desde el principio, fue la apuesta por herramientas opensource. Basándonos en el estudio del estado del arte y en las comparativas realizadas entre diferentes productos, finalmente nos decantamos por las siguientes:

- Hardware físico:

  El entorno se creó sobre ordenadores de uno de los laboratorios de Ericsson España S.A.. El laboratorio utilizado está compuesto de un cluster de 6 máquinas con las siguientes características de procesador y memoria:

  - CPU: Intel Core 2 Quad Q6600 2.40GHz
  - Memoria RAM: 8GB

  Entendemos que tanto el número de equipos como sus especificaciones son suficientes para nuestros propósitos.

- Sistema operativo anfitrión:

  El sistema operativo utilizado fue Ubuntu 9.04 Jaunty Jackalope. Nos decantamos por este sistema operativo, ya que necesitábamos un sistema UNIX. Además, esta versión de Ubuntu apuesta por el Cloud Computing incluyendo entre otras herramientas KVM, Eucalyptus y OpenNebula.

- Hypervisor:

  Este apartado fue uno de los que más dificultades nos ocasionó. Actualmente existen varios hypervisores con buena reputación en el mercado (en el apéndice A puede encontrarse más información sobre este tipo de herramientas). Principalmente, las dudas que nos surgieron se repartieron entre Xen y KVM, ya que ambos son opensource y son soportados por la mayoría de los gestores de infraestructura virtual. A pesar de que Xen es más veterano y utilizado a día de hoy, nuestra elección final fue KVM por las siguientes razones primordialmente:

    - Es más sencillo de configurar. Entre otras cosas no es necesario modificar el gestor de arranque GRUB ni el sistema operativo huésped (tampoco sería necesario con Xen utilizando virtualización completa).

  - Ubuntu apuesta por KVM como su principal solución de virtualización.
  - Es la solución opensource más prometedora según muchos tecnólogos.

- Gestor de infraestructura virtual:

  El gestor de la infraestructura virtual es el núcleo para la construcción de nuestro entorno cloud. Como vimos en capítulos anteriores, existen algunas alternativas en este apartado. De entre todas ellas, al final nos decidimos por OpenNebula al ser uno de los más completos y que más funcionalidades ofrece. Además, tiene la ventaja de ser opensource y de ofrecer la posibilidad de construir cualquiera de los 3 tipos de nubes (públicas, privadas e híbridas). Cabe destacar que está desarrollado por el grupo de investigación de sistemas distribuidos (dsa-research) de la Universidad Complutense de Madrid.

  La versión utilizada en este proyecto es OpenNebula 1.2, ya que en la fecha en la que se inició aún no se había liberado la la versión 1.4.

- Base de datos distribuida:

  La base de datos elegida fue Mnesia. Es una BD distribuida escrita en Erlang. Proporciona primitivas para añadir nodos a la BD y dividir las tablas en fragmentos. Esta es la razón por la que nos decantamos por esta herramienta, que además es opensource.

  En los apéndices B y C se puede encontrar más información sobre Erlang y Mnesia.

4.3 Arquitectura

Una vez tuvimos claras todas las piezas necesarias para desarrollar nuestro sistema, ya sólo quedaba combinarlas de la manera correcta. La arquitectura propuesta puede verse en el diagrama siguiente:

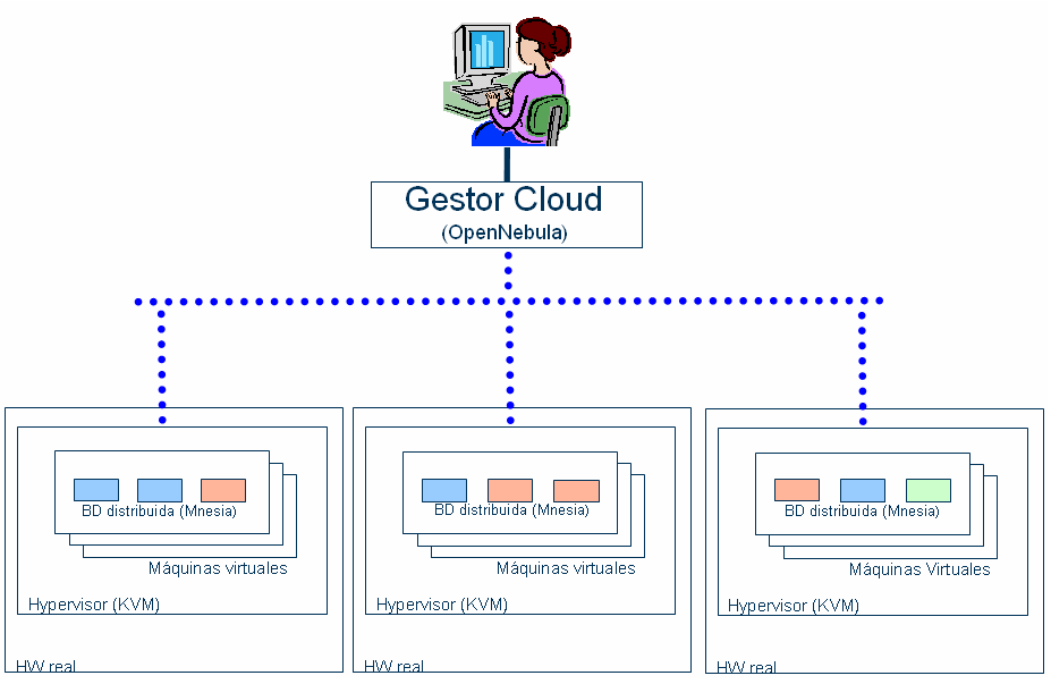

Figura 40 Arquitectura del entorno cloud

4.4 Construcción del entorno

Teniendo claros los dos puntos anteriores, ya sólo quedaba levantar el entorno en el laboratorio. A continuación, se muestran los pasos seguidos para llevar a cabo esta tarea:

1) Máquinas físicas:

   Cómo máquina física anfitriona utilizamos uno de los ordenadores del laboratorio con las características anteriormente descritas. Como ya se ha mencionado, el sistema operativo instalado en esta máquina es Ubuntu 9.04.

2) Instalación del hypervisor:

   Antes de la instalación de KVM, fue necesario instalar los siguientes paquetes:

   - Libvirt: Herramienta que proporciona un API para operaciones de virtualización.
   - Ubuntu-vm-builder: Utilidad para creación de máquinas virtuales personalizadas.

- Qemu: Emulador de procesadores basado en la traducción dinámica de binarios (conversión del código binario de la arquitectura fuente en código entendible por la arquitectura huésped). También tiene capacidades de virtualización dentro de un sistema operativo.
- Bridge-utils: Utilidad para la creación de bridges en Linux.
- Virsh: Interfaz para la gestión de dominios huésped. Puede utilizarse para cread, pausar y apagar dominios. Depende de la librería libvirt.
- Virt-viewer: Herramienta que muestra las máquinas virtuales instanciadas a través de una consola.

Cuando todos estos paquetes estuvieron instalados, el siguiente paso fue la instalación de KVM. Para ello seguimos los pasos descritos en los siguientes manuales:

- https://help.ubuntu.com/community/KVM
- http://www.howtoforge.com/virtualization-with-kvm-on-ubuntu-9.04

Creamos cuatro máquinas virtuales con una unidad de CPU y 512 MB de RAM con sistema operativo Ubuntu 8.10. Para la red, se creó un bridge DHCP en la máquina anfitriona.

3) Instalación del gestor de la infraestructura virtual:

Para la instalación del software del Front-End y de los nodos de OpenNebula seguimos como referencia los siguientes manuales:

- http://www.opennebula.org
- http://doc.ubuntu.com/ubuntu/serverguide/C/opennebula.html
- https://help.ubuntu.com/community/OpenNebula

Una vez instalado, OpenNebula necesita ser configurado. En primer lugar, creamos un nodo (onehost create). Después, diseñamos una plantilla con la configuracón de la red virtual donde le indicamos el nombre del bridge configurado anteriormente y la añadimos al gestor (onevnet create). Finalmente, es necesario elaborar otra plantilla con las características de cada máquina virtual. En este punto, ya es posible arrancar nuestras máquinas virtuales utilizando OpenNebula.

4) Instalación de la BD:

Para la instalación de Mnesia, basta simplemente con descargar el entorno OTP desde su página web e instalarlo. Una vez instalado, es posible utilizar todas las herramientas ofrecidas por el lenguaje Erlang, incluyendo su base de datos Mnesia.

5 Pruebas y resultados obtenidos

En este capítulo, se muestran los resultados obtenidos al evaluar el comportamiento de la base de datos distribuida Mnesia sobre el entorno cloud construido. Además, comparamos el rendimiento, con el alcanzado al realizar esas mismas pruebas sobre sobre un entorno tradicional y sobre una única máquina virtual arrancada directamente con el hypervisor.

La prueba realizada consistió en la creación de una tabla de empleados (employee) con dos atributos, identificador (id) y salario (salary).

id	salary

Figura 41 Tabla employee

Una vez creada, introdujimos un número elevado de registros utilizando la función write ofrecida por Mnesia. La función diseñada para la realización de esta tarea es la siguiente:

```
insert(N) ->
 Write = fun(Keys) ->
    [mnesia:write({employee,K,K+1000}) || K <- Keys],
    ok
 end,
 mnesia:activity(sync_dirty,Write,[lists:seq(1,N)],mnesia_frag).
```

Esta función recibe un argumento que le indica el número de registros a introducir en la tabla. En nuestro caso fueron 1000000 en todos los casos.

Una vez rellena la tabla, vamos extrayendo un número de registros creciente (comenzando en 50000 y aumentando 50000 en cada nueva extracción hasta llegar a 1000000) y registrando los tiempos. De esta forma, es posible comparar el rendimiento/latencia de la BD en las diferentes situaciones propuestas y extraer conclusiones acerca de cómo afecta el entorno cloud en su ejecución. La función creada para este propósito se muestra debajo:

```
read_n_employees(N) ->
 Read = fun(Key) ->
    mnesia:read({employee, Key})
 end,
 Before = now(),
 mnesia:activity(transaction,Read,[lists:seq(1,N)],mnesia_frag),
 After = now(),
 {Before,After}.
```

El parámetro N indica el número de registros a leer en cada llamada.

En todos los casos, la configuración de Mnesia fue la siguiente:

- Esquema (schema): Disc copies (todos los datos se mantienen en memoria y cada vez que se realiza una actualización en la tabla se refleja en el disco).
- Tabla empleado (employee): Ram copies (los datos son almacenados en memoria únicamente).

Además, las llamadas a las funciones no las hacíamos directamente desde las máquinas en que se estaba ejecutando la BD. De esta forma, evitamos introducir más carga para que los tiempos obtenidos fuesen más precisos. Para ello, nos conectábamos desde la consola de Erlang de una máquina remota a una de las máquinas en donde estaba corriendo la DB y realizábamos las llamadas necesarias.

En los apartados contiguos se muestran los resultados obtenidos en los distintos casos.

5.1 Ejecución de la BD sobre hardware tradicional

En primer lugar, realizamos la prueba descrita en el apartado anterior sobre un computador físico tradicional de los existentes en el laboratorio que cuenta con las siguientes características:

- CPU: Intel Core 2 Quad Q6600 2.40GHz
- Memoria RAM: 8GB
- Sistema operative: Ubuntu 9.04

Los resultados obtenidos fueron los siguientes:

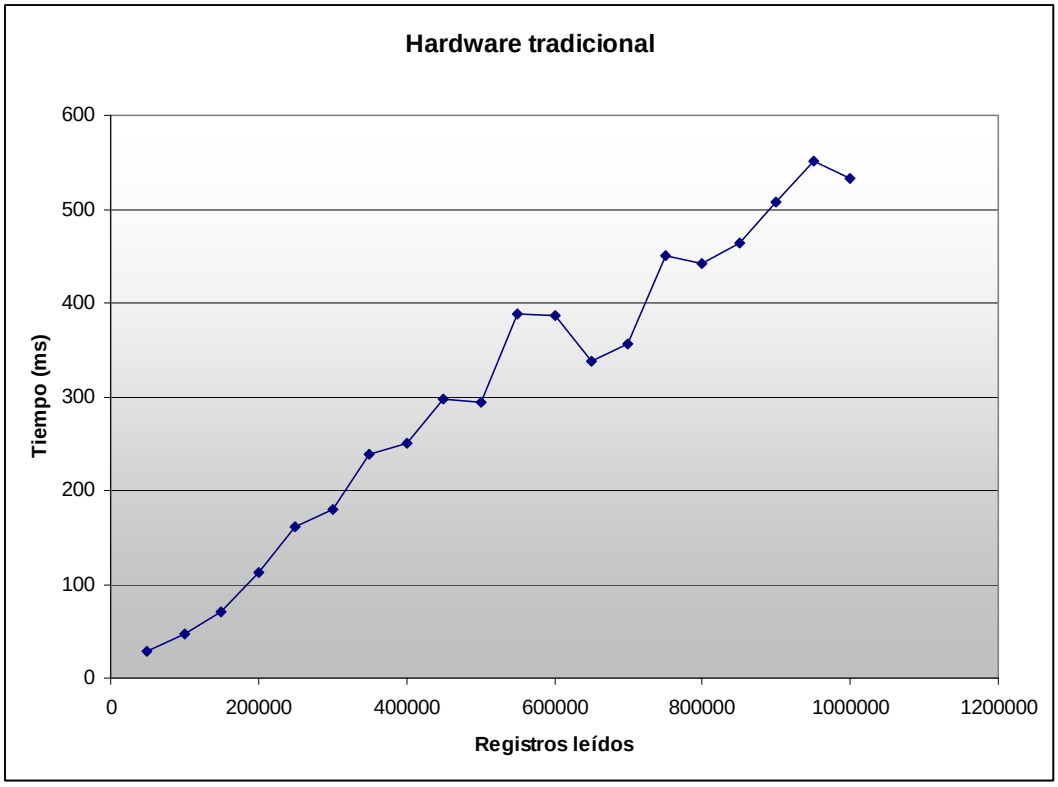

Figura 42 Ejecución de Mnesia sobre hardware tradicional

El eje de abcisas muestra el número de registros leídos, mientras que el eje de ordenadas muestra los tiempos consumidos para leer estos registros en milisegundos.

5.2 Ejecución de la BD sobre una máquina virtual

Para realizar este apartado, arrancamos una máquina virtual mediante KVM, sin la intervención de OpenNebula y ejecutamos la prueba explicada al principio del capítulo. Los resultados obtenidos se muestran en la siguiente gráfica:

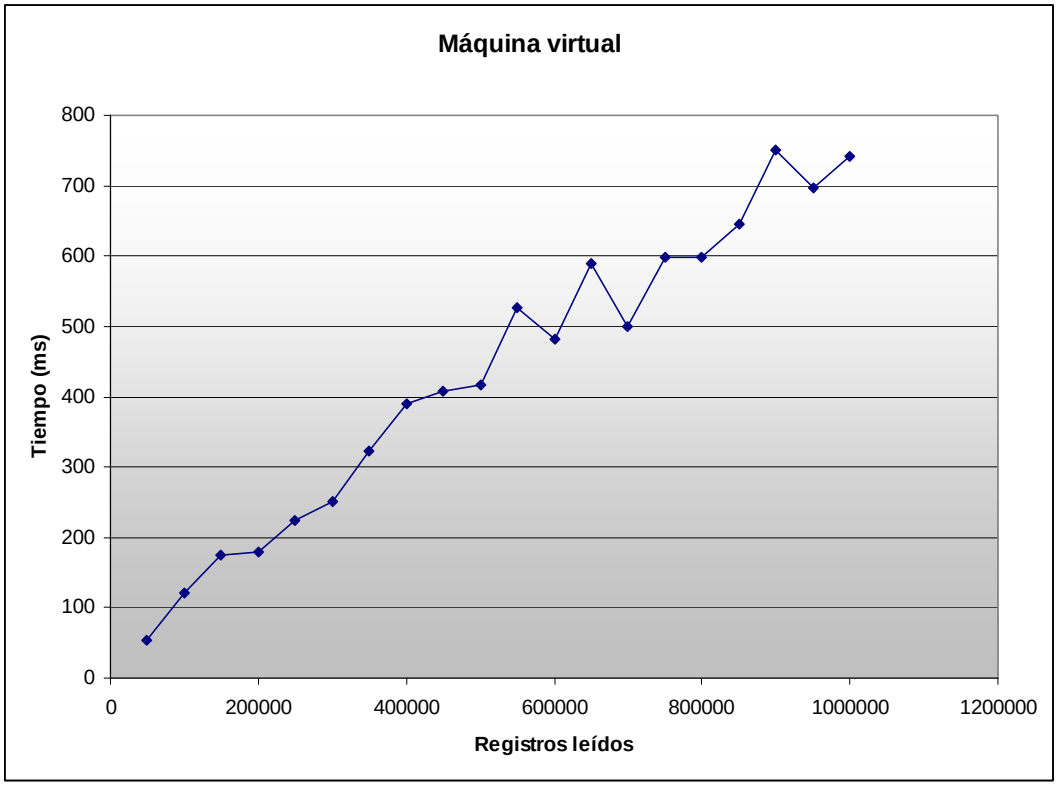

Figura 43 Ejecución de Mnesia sobre una máquina virtual

5.3 Ejecución de la BD sobre un cloud privado

Finalmente, las últimas pruebas las realizamos sobre el cloud privado construido en el laboratorio. Una de las principales características de los entornos cloud, es que son elásticos. Para poder apreciar esta capacidad, al principio comenzamos con una sola máquina virtual, y fuimos desplegando más máquinas dinámicamente hasta un total de cuatro, de forma que se podía observar cómo la tabla de empleados se iba redistribuyendo según se añadían nuevos nodos en caliente a la BD. A continuación, podemos ver los resultados obtenidos en cada fase.

1) BD ejecutándose en un nodo:

   Seguidamente, mostramos unas capturas donde se puede ver Mnesia ejecutándose en una única máquina virtual (vmone1).

También se aprecia, que el número de registros almacenados en la tabla empleados es de un millón, localizados en un único fragmento.

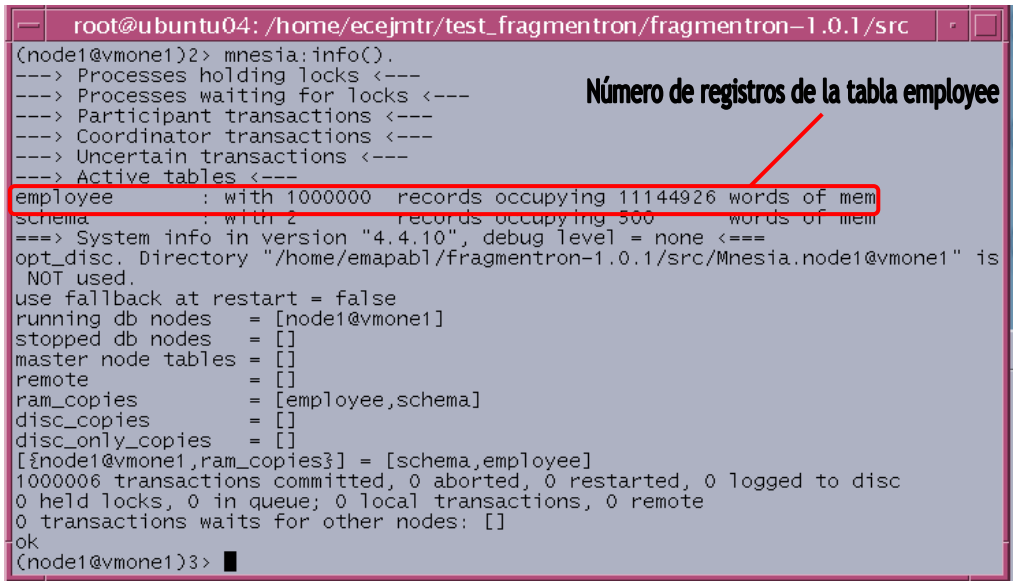

Figura 44 BD Mnesia desplegada sobre un nodo (I)

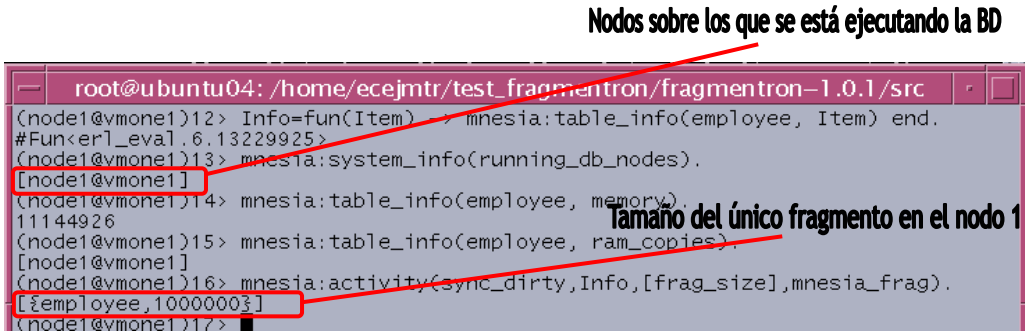

Figura 45 BD Mnesia desplegada sobre un nodo (II)

En este caso, no anotamos los tiempos requeridos para extraer registros de la tabla employee, ya que consideramos que los resultados serían prácticamente los mismos que los obtenidos en el apartado 5.2

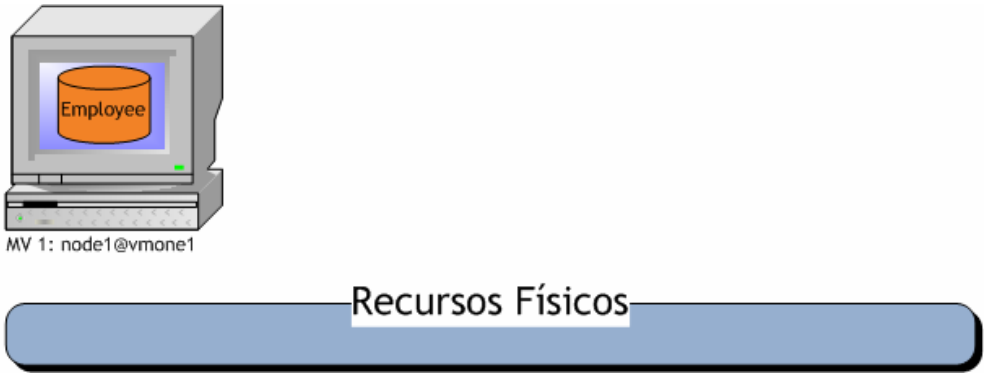

Figura 46 Tabla Employee en la MV 1

2) BD ejecutándose en dos nodos:

Al desplegar el segundo nodo, informamos a la base de datos de la existencia de un nuevo nodo y de que debía dividir la tabla de empleados en fragmentos para distribuir los registros entre el total de nodos disponibles. En la figura posterior, se puede apreciar toda esta información. Además, se observa como el número de registros repartidos en cada fragmento es prácticamente el mismo.

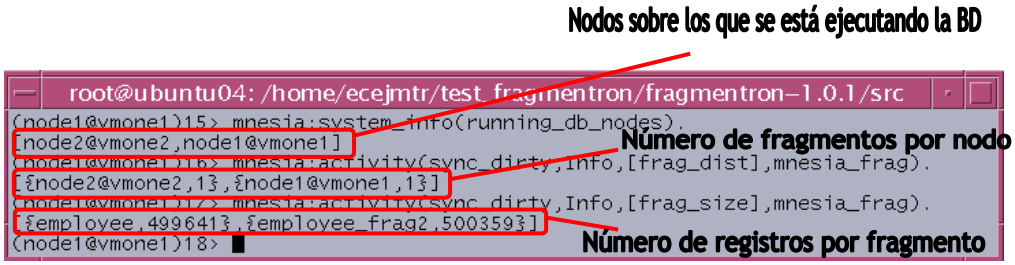

Figura 47 BD Mnesia desplegada sobre dos nodos

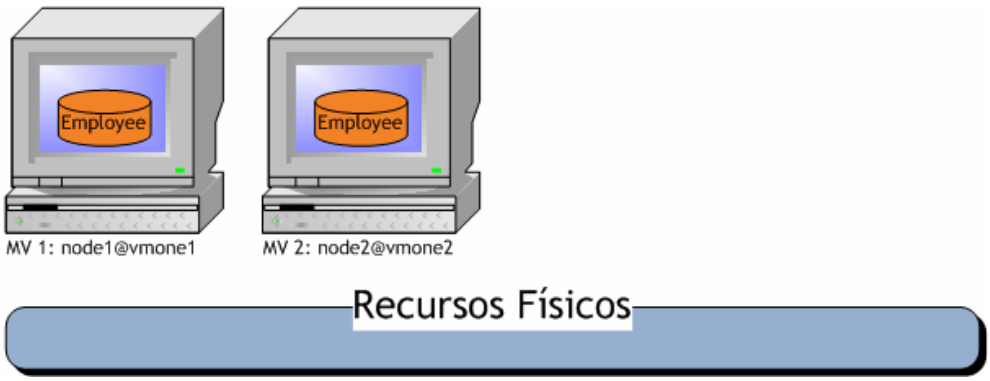

Figura 48 Tabla Employee dividida en 2 fragmentos

A continuación, ejecutamos las mismas pruebas que en los apartados anteriores, obteniendo los siguientes resultados:

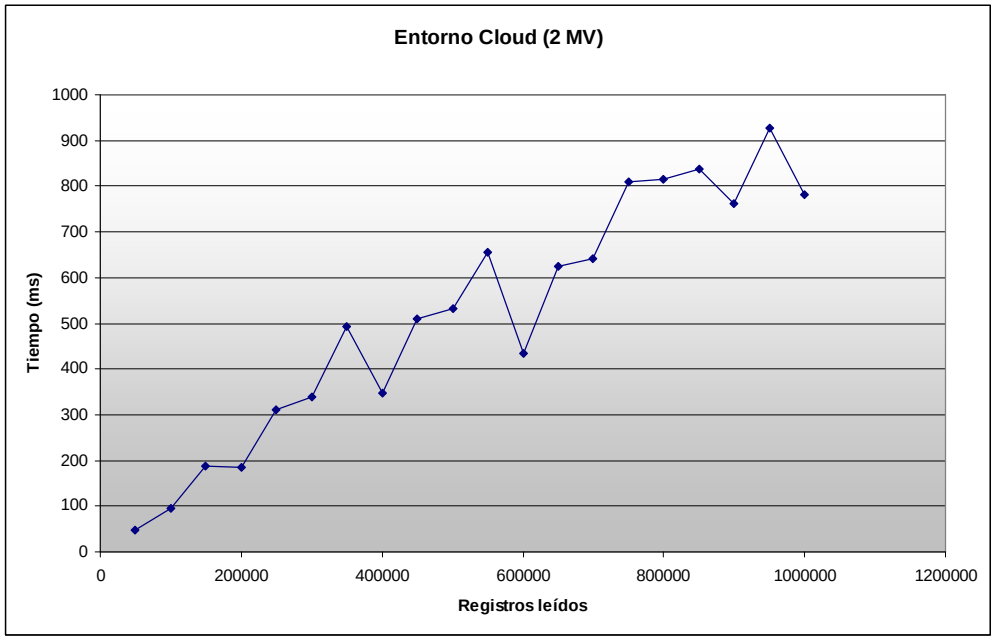

Figura 49 Ejecución de Mnesia sobre dos nodos

3) DB ejecutándose en 3 nodos:

El paso siguiente, fue arrancar una nueva máquina virtual y añadirla en la configuración de Mnesia. En la siguiente figura se puede ver que no hubo ningún problema y un nuevo fragmento fue añadido a la tabla y ubicado en el nuevo nodo:

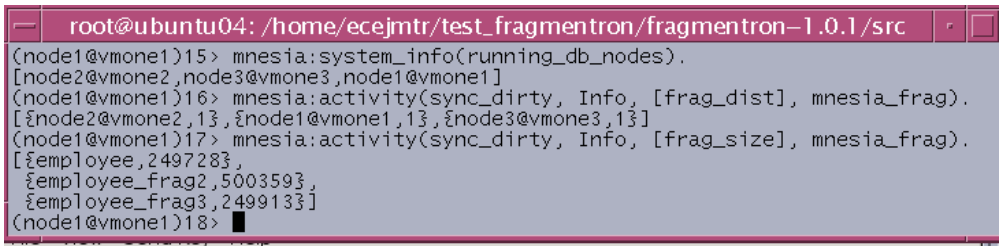

Figura 50 Mnesia desplegada sobre 3 nodos

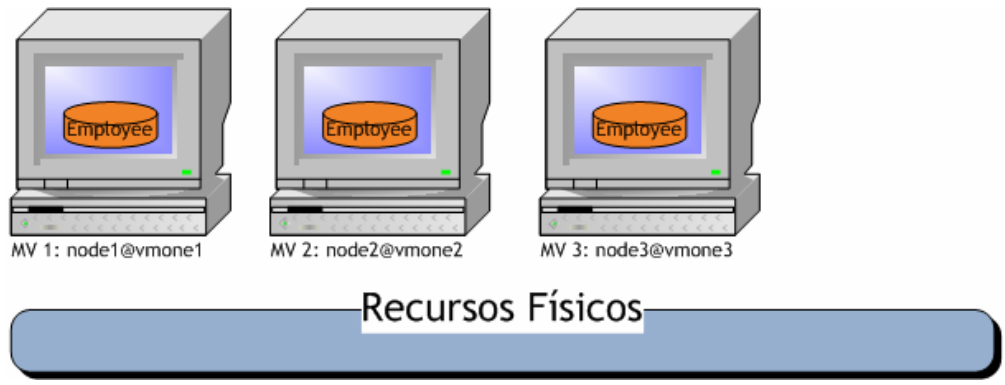

Figura 51 Tabla Employee dividida en 3 fragmentos

Los tiempos obtenidos al realizar la lectura de registros en este caso se pueden ver en la figura siguiente:

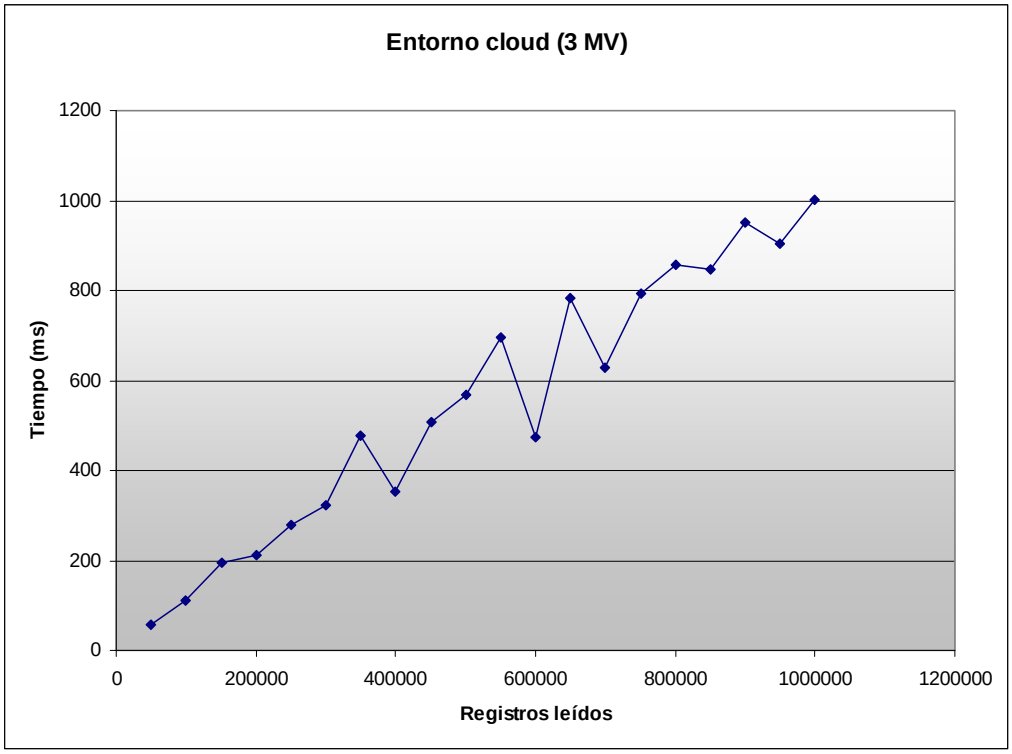

Figura 52 Ejecución de Mnesia sobre tres nodos

4) BD ejecutándose en 4 nodos:

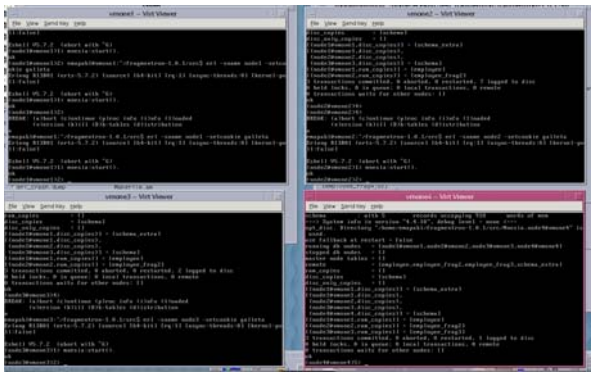

Figura 53 Máquinas virtuales desplegadas con OpenNebula

Por último, una cuarta máquina virtual fue levantada y añadida al sistema:

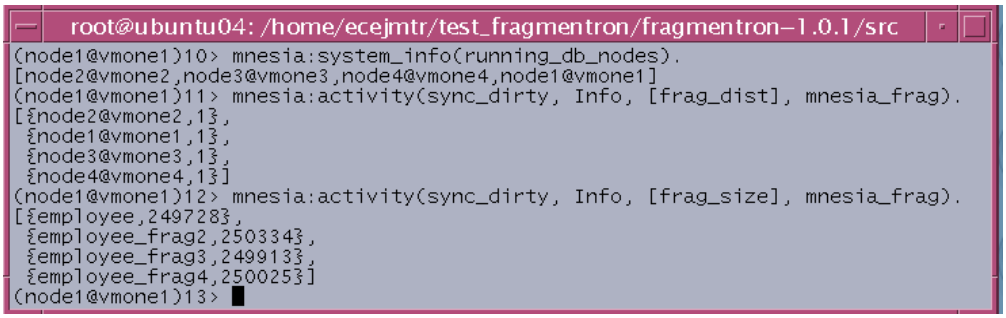

Figura 54 Mnesia desplegada sobre 4 nodos

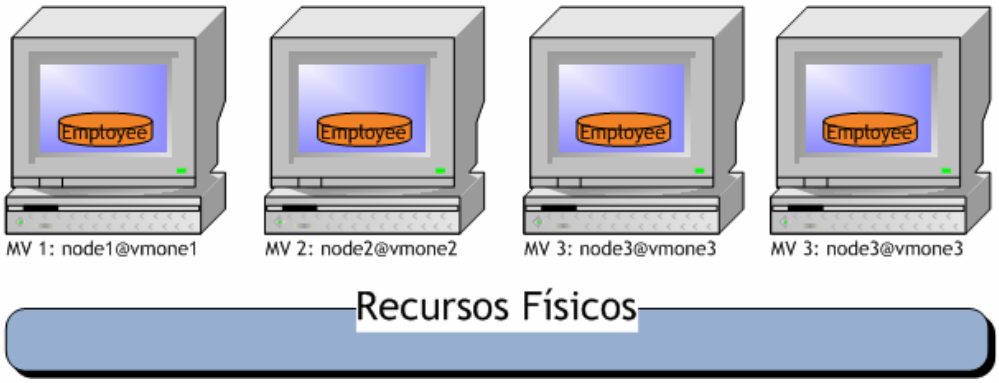

Figura 55 Tabla Employee dividida en 4 fragmentos

Los resultados obtenidos en este último caso después de realizar el estudio de tiempos se encuentran en la página siguiente.

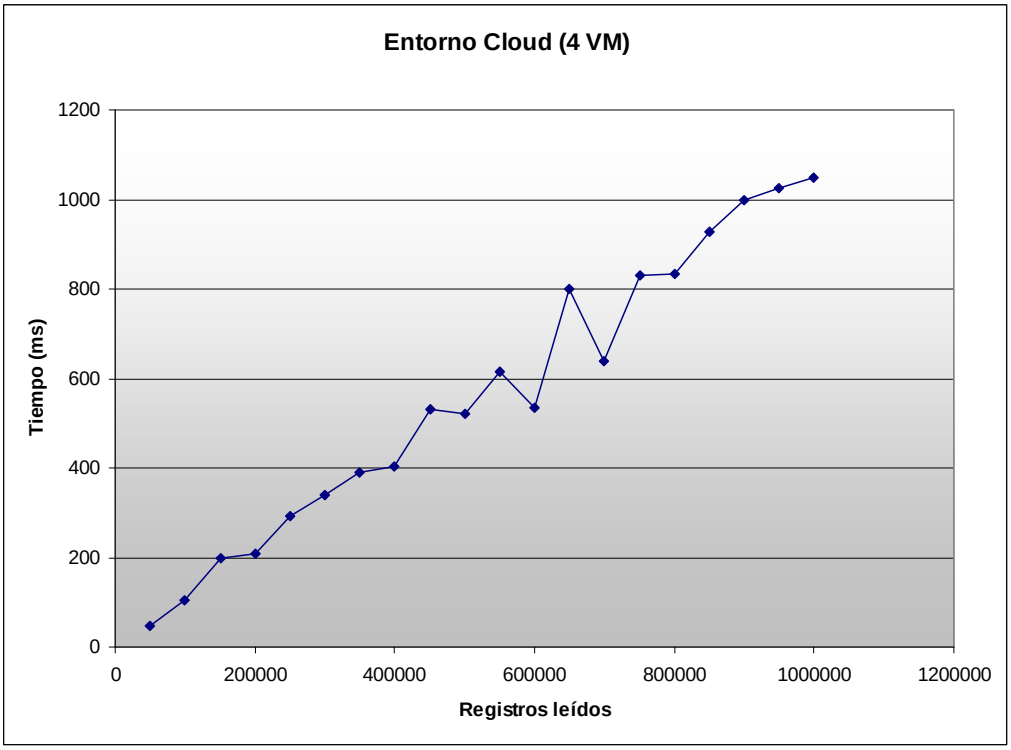

Figura 56 Ejecución de Mnesia sobre cuatro nodos

5.4 Comparativa de los resultados obtenidos

Después de realizar la totalidad de las pruebas anteriores y conocer los resultados obtenidos en cada una de ellas, se elaboró una gráfica comparativa, donde se pueden apreciar las diferencias entre los distintos casos de forma sencilla y visual.

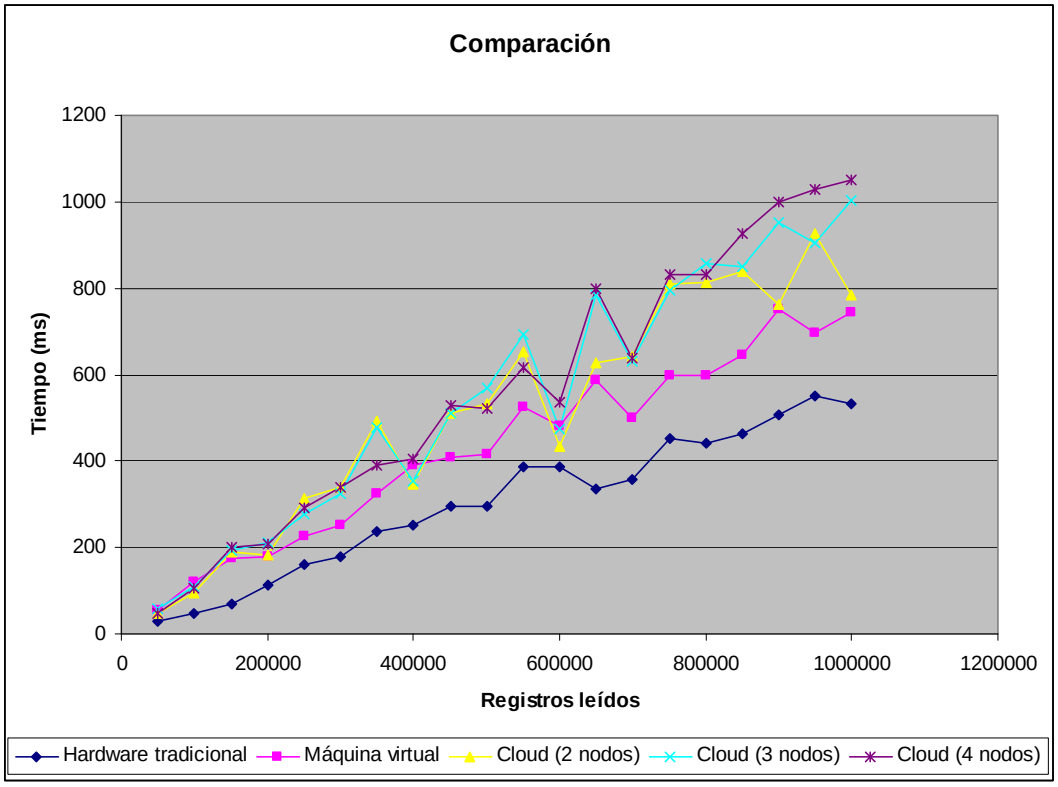

Figura 57 Comparativa entre distintas ejecuciones

Como es lógico, la máquina tradicional es la que obtiene los mejores resultados, ya que se ejecuta la BD directamente sobre el hardware físico, sin ninguna capa intermedia que influya en el rendimiento. En segundo lugar aparece la máquina virtual. Se observa claramente que los tiempos de acceso son más elevados. Sin embargo, la diferencia obtenida no es demasiado alta y está en torno al 40-45 % de aumento. El entorno cloud aparece en tercer lugar, y los tiempos obtenidos aumentan según se van añadiendo nodos. De todas formas, este aumento no es demasiado elevado, ya que según añadimos nuevos nodos, el rendimiento empeora sólo entre un 5 % y un 10 %. Además, cuanto menor es el número de registros leídos, los tiempos son más parejos. Para finalizar, cabe destacar, que el aumento de tiempo entre la máquina tradicional y la infraestructura virtual de 4 nodos se encuentra situado entre el 90 % y el 100 %.

6 Conclusiones

Al comenzar este proyecto fue necesario un gran trabajo de investigación sobre este modelo de computación conocido como Cloud Computing. Como se ha mencionado en alguna ocasión en este trabajo, Cloud Computing es un término bastante confuso en la actualidad, incluso para muchas personas relacionadas con el mundo de la computación. Es por ello, que en un primer momento, las dudas sobre lo que este término significaba exactamente eran demasiadas. Sin embargo, una vez llegados a este punto y después de haber desechado una gran cantidad de información superflua, creemos que se ha conseguido plasmar en estas páginas una idea clara del concepto y una visión general sobre este paradigma, bastante ajustada a la realidad.

La construcción de un cloud privado y el posterior despliegue sobre él de una base de datos distribuida como Mnesia ha sido también de gran ayuda para extraer algunas conclusiones sobre este tipo de arquitecturas.

En primer lugar, es importante destacar el beneficio que supone tener una arquitectura elástica y escalable bajo demanda. Por un lado, es posible aumentar nuestra capacidad de procesamiento y almacenamiento de una forma totalmente dinámica y elástica y aparentemente sin límites. Esto es posible conseguirlo incluso en cloud privados como el montado en este proyecto, ya que si en algún momento nuestra infraestructura se ve saturada, se pueden contratar nuevas máquinas virtuales a través de Internet a un proveedor externo, obteniendo lo que se conoce como cloud híbrido y siendo capaces de responder a demandas pico en minutos. Por otro lado, es importante añadir, que la utilización de los recursos disponibles es más óptima siguiendo este modelo, ya que en todo momento sólo utilizamos los recursos que necesitamos. Este dato es bastante importante, ya que puede repercutir en una reducción de costes y en un modelo de computación mucho más ecológica y respetuosa con el medio ambiente (Green Computing).

Uno de los puntos débiles de este tipo de arquitecturas, es el aumento de la latencia debido a la insuficiente velocidad de las conexiones a internet. Esta cuestión no es tan importante en el caso de nubes privadas como la descrita en los últimos capítulos de esta memoria, ya que la infraestructura se encuentra dentro de la propia empresa. Sin embargo, es importante destacar que tanto en el caso de nubes públicas como privadas existe una disminución en el rendimiento (no demasiado) de nuestras aplicaciones. Esto es debido principalmente a la inclusión de la capa de virtualización y de gestión de de la infraestructura. Además, también hay que tener en cuenta, que varias máquinas virtuales se están ejecutando en la misma máquina física y todas comparten las mismas interfaces para entrada/salida, lo que supone un cuello de botella para aquellas aplicaciones que hagan un uso elevado de esta característica.

Por tanto, queda claro que el cloud no es un lugar apropiado para sistemas altamente transaccionales o datos sensibles a latencia. Sin embargo, para otro tipo de aplicaciones, en que estos requisitos no sean tan extrictos, el cloud puede proporcionar multitud de beneficios como los citados en las páginas anteriores.

Por último, añadir unas ideas sobre trabajo futuro. En primer lugar, sería interesante, realizar pruebas similares a las descritas aquí en un mayor número de máquinas físicas y virtuales y ver hasta que punto es capaz de extenderse la infraestructura sin problemas. También sería bueno, intentar acceder a un cloud público para extender los recursos disponibles y extraer conclusiones de cómo influiría en el rendiminento y la latencia del sistema. Otro punto a tener en cuenta es que pasaría si mientras se están redistribuyendo los fragmentos de una tabla hacemos una consulta. Incluso realizar esta misma consulta mientras se está migrando una máquina virtual de una máquina física a otra. Como vemos, las posibilidades dentro de este campo son infinitas, ya que el Cloud Computing aún está en una fase de crecimiento y asentamiento.

Apéndice A – Hypervisores

Un hypervisor, también llamado Virtual Machine Monitor (VMM), es un programa que permite que múltiples sistemas operativos compartan un único hardware de forma aislada y al mismo tiempo. Se comunica con el sistema anfitrión, ya sea hardware o sistema operativo, para interoperar entre las máquinas virtuales y el sistema físico.

El VMM crea una capa de abstracción entre el hardware de la máquina física (anfitrión o host) y el sistema operativo de la máquina virtual (invitado, huesped o guest), de tal forma que maneja los recursos de las máquinas físicas subyacentes de una manera que el usuario pueda crear varias máquinas virtuales presentando a cada una de ellas una interfaz del hardware que sea compatible con el sistema operativo elegido.

Esta capa de software maneja, gestiona y arbitra los cuatro recursos principales de una computadora (CPU, memoria, red y almacenamiento) de forma que puede repartir dinámicamente dichos recursos entre todas las máquinas virtuales definidas en el computador central.

En la actualidad, todos los fabricantes tanto de software como de hardware están trabajando para mejorar, ayudar al Hypervisor (VMM) y asi poder llegar a una virtualización completa, fiable y robusta.

Existen varios Tipos de Hypervisor:

- Tipo 1 (nativo, baremetal o unhosted): Este tipo de hypervisor opera como una capa intermedia entre el hardware y los sistemas operativos huéspedes. Todas las traducciones binarias (binary translation) de CPU, memoria, red y almacenamiento las hace la capa VMM.

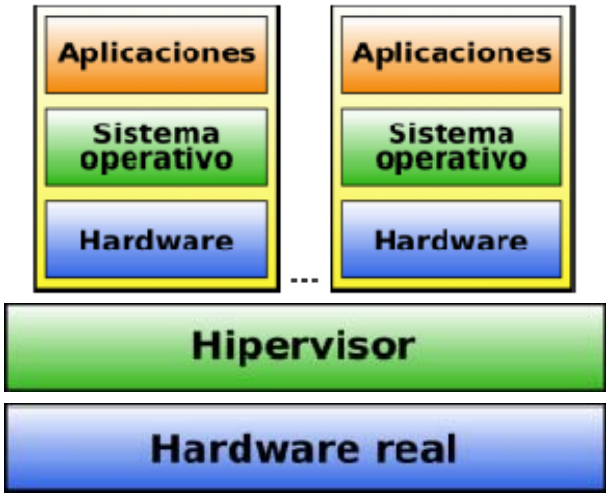

Figura 58 Hypervisores tipo 1

Algunos de los hipervisores tipo 1 más destacados son los siguientes:

- VMware: ESXi (gratis), ESX (de pago)
- Xen (libre)
- Citrix XenServer (gratis)
- Microsoft Hyper-V Server (gratis)

- Tipo 2 (hosted): Este tipo de hypervisor necesita un sistema operativo completo para poder ejecutarse. De esta forma la virtualización se produce en una capa más alejada del hardware si lo comparamos con los hipervisores de tipo 1. Lógicamente esto hace que el rendimiento sea menor en los hipervisores de tipo 2.

Figura 59 Hypervisores tipo 2

Algunos de los hipervisores tipo 2 más utilizados son los siguientes:

- Sun: VirtualBox (gratis)
- VMware: Workstation (de pago), Server (gratis), Player (gratis)
- QEMU (libre)
- KVM (libre)
- Micorsoft: Virtual PC, Virtual Server

Apéndice B – Erlang

Erlang es un lenguaje de programación declarativo, concurrente y distribuido orientado a aplicaciones de telecomunicaciones. Es también un sistema de ejecución que incluye una máquina virtual y bibliotecas. La distribución del lenguaje junto con las librerías y la base de datos de tiempo real (Mnesia) se denominan Open Telecom Platform (Plataforma abierta de Telecomunicaciones), u OTP. Tiene muchas características comunmente asociadas con un sistema operativo: concurrencia de procesos, planificación, gestión de memoria, red, etc. Fue diseñado en la compañía Ericsson. Originalmente era un lenguaje propietario de esta empresa, pero fue cedido como open source en 1998. Erlang recibe el nombre de A. K. Erlang. A veces se piensa que el nombre es una abreviación de ERicsson LANGuage, debido a su uso intensivo en Ericsson. Según Bjarne Däcker quién fue el jefe del Computer Science Lab en su día, esta dualidad es intencional.

Erlang pertenece a la clase de lenguajes orientados a mensajes. Estos lenguajes proporcionan concurrencia usando procesos paralelos. No hay objetos compartidos en un lenguaje orientado a mensajes, sino que toda la interacción entre los procesos se realiza enviando y recibiendo mensajes. El modelo de procesos se basa en el modelo de comunicación de procesos secuenciales ideado por Sir Charles Antony Richard Hoare.

Sus características más destacadas son:

- Lenguaje funcional
- Soporte de concurrencia
- Distribución, no hay memoria compartida
- Robustez y tolerancia a fallos (tiene varias primitivas de detección de errores)
- Soporta programación de sistemas de tiempo real
- Recolector de basura
- Reemplazo de código en caliente
- Procesos concurrentes altamente aislados
- Comunicación entre procesos con paso de mensajes
- Funciones de orden superior
- Comprobación de tipos dinámica
- Evaluación de argumentos estricta (temprana, para permitir efectos secundarios)
- Posibilidad de conectar con código en C, Java y otros lenguajes

Respecto a la concurrencia:

- Todo es un proceso
- Los procesos están fuertemente aislados
- La creación y destrucción de objetos es un proceso ligero
- La única forma en que los procesos pueden interaccionar es con el paso de mensajes
- Los procesos tienen nombres únicos
- Si conoces el nombre de un proceso puedes mandarle un mensaje
- Los procesos no comparten recurso alguno
- El manejo de errores no es local
- Los procesos hacen lo que se supone que han de hacer, o fallan

Todas estas características fueron desarrolladas por el equipo de Erlang teniendo en cuenta los requisitos que tienen las aplicaciones de telecomunicaciones: alta disponibilidad, modificación del código sin detener la aplicación, etc.

Un servidor en Erlang normalmente se implementa como una pequeña función, con argumentos de entrada que representan el estado del servidor. El cuerpo de la función recibe un mensaje, realiza la computación requerida, envía el resultado y hace una llamada en la cola a sí mismo con parámetros que representen el nuevo estado. De esta forma, basta lanzar varios procesos con esa función en las máquinas de un cluster. También simplifica el reemplazo de código en caliente, ya que no hay un hilo de código que se ejecute permanentemente, sino que la función principal se va llamando a sí misma recursivamente.

La mayor parte del lenguaje es el subconjunto secuencial. Este subconjunto secuencial tiene comprobación dinámica de tipos, y es un lenguaje de programación estrictamente funcional, libre de efectos secundarios en su mayor parte. En este subconjunto secuencial hay unas pocas operaciones con efectos secundarios, pero apenas se usan.

Los programas en Erlang están compuestos de bloques funcionales, llamados funciones. Las funciones se identifican unívocamente por su nombre y su aridad (el número de datos de entrada), no solo por su nombre como en otros lenguajes. Por tanto, las funciones suma/2 y suma/3 que aceptan 2 y 3 datos de entrada respectivamente son distintas y pueden coexistir perfectamente. Las funciones devuelven siempre un resultado, que puede ser un elemento Erlang válido como por ejemplo un entero, un flotante, una lista, una tupla, etc. Si se desean devolver varios datos, lo normal es componer una tupla con ellos.

Erlang viene varios componentes que pueden ser utilizados como bloques de construcción en el desarrollo de aplicaciones. Entre ellos cabe destacar Inets, Mnesia, Orber y SNMP.

Algunas librerías y herramientas incluidas:

- ODBC: Proporciona una interfaz para acceder a bases de datos relacionales SQL.
- Crypto y ssl: Proporciona funciones para computar resúmenes de mensajes, cifrado y descifrado. Usa partes de OpenSSL.
- Appmon: Monitorización gráfica de grupos de procesos.
- ASN.1: Paquete de compilación y ejecución con soporte para ASN.1.
- Compiler: Compilador.
- Debugger: Depurador gráfico.
- ERTS: Entorno de ejecución de Erlang. Incluye la máquina virtual, el recolector de basura, etc.
- GS: Librería para escribir interfaces gráficas de usuario.
- IC: Compilador de OMG's IDL (Interface Definition Language) a Erlang, C y Java.
- Kernel: Código C necesario para ejecutar el sistema Erlang.
- Mnemosyne: Lenguaje de consultas opcional para Mnesia.
- Mnesia Session: Interfaz exterior hacia Mnesia definida en IDL.
- OS monitor (OS_MON): Monitorización de CPU, disco y memoria.
- PMan: Herramienta para trazar y observer el estado de los procesos de Erlang (localmente o desde nodos remotos).
- Stdlib: Librerías para entrada/salida, gráficos, listas, conjuntos, colas, expresiones regulares, etc.
- Table visualizer: Herramienta para visualizar ETS y tables de Mnesia.
- Toolbar: Barra de herramientas que simplifica el acceso a las herramientas de Erlang.
- Otras herramientas.

Apéndice C - Mnesia

Mnesia es un sistema de gestión de bases de datos (Data Base Management System o DBMS) distribuido, apropiado para aplicaciones de telecomunicaciones. Es un componente del Open Telecom Platform (OTP).

La gestión de datos en sistemas de telecomunicaciones tiene muchos aspectos de los que, algunos, aunque no todos, son abordados por los DBMS comerciales tradicionales. En particular, el alto nivel de tolerancia a fallos que se requiere en muchos sistemas "nonstop", junto con los requisitos sobre el DBMS para ejecutarse en el mismo espacio de direcciones que la aplicación, llevaron a los creadores de Mnesia a implementar un nuevo tipo de BD. Mnesia está implementado con el lenguaje de programación Erlang y proporciona la funcionalidad necesaria para la implementación de sistemas de telecomunicaciones altamente tolerantes a fallos. Mnesia trata de abordar todos los temas de gestión de datos requeridos por los sistemas de telecomunicaciones típicos y tiene un número considerable de características que normalmente no se encuentran en bases de datos tradicionales.
En las aplicaciones de telecomunicaciones existen diferentes necesidades que hacían necesaria la creación de un DBMS de este estilo. Mnesia está diseñado con requisitos como los siguientes en mente:

- Búsquedas clave/valor rápidas (en tiempo real)
- Complicadas consultas (no de tiempo real) principalmente para operaciones de mantenimiento
- Datos distribuidos debido a aplicaciones distribuidas
- Alto nivel de tolerancia a fallos
- Reconfiguración dinámica
- Objetos complejos

La diferencia entre Mnesia y la mayoría de los DBMS clásicos es que está diseñado con los problemas típicos de la gestión de datos en aplicaciones de telecomunicaciones en mente. Por tanto, Mnesia combina muchos conceptos que se encuentran en bases de datos típicas, tales como transacciones y consultas, con conceptos como rápidas operaciones en tiempo real, grado de tolerancia a fallos configurable (a través de replicación) y capacidad para reconfigurar el sistema sin la necesidad de pararlo o suspenderlo. Mnesia también es interesante debido a su estrecho acoplamiento con el lenguaje de programación Erlang, convirtiéndolo casi en un lenguaje de programación de bases de datos. Esto tiene muchos beneficios, el más importante es que la falta de concordancia entre el formato de datos utilizado por el DBMS y el formato de datos utilizado por el lenguaje de programación, que se utiliza para manipular los datos, desaparece por completo.

Algunas de sus principales características son:

- El esquema de la BD puede ser reconfigurado dinámicamente en tiempo de ejecución.
- Las tablas pueden ser declaradas con propiedades tales como localización, replicación y persistencia.
- Las tablas pueden ser movidas o replicadas a varios nodos para mejorar la tolerancia a fallos.
- La localización de las tablas es transparente al programador.
- Las transacciones de la BD pueden ser distribuidas y un gran número de funciones pueden ser llamadas dentro de una transacción.
- Varias transacciones pueden ser ejecutadas concurrentemente y su ejecución está completamente sincronizada por el DBMS. Mnesia asegura que no habrá dos procesos que manipularán los datos simultáneamente.
- A las transacciones se les puede asignar la propiedad de ser ejecutadas sobre todos los nodos del sistema o en ninguno.

Bibliografía

- Libros:
  - CARR, Nicholas. *The Big Switch: Rewiring the world, from Edison to Google*. Nueva York: W.W. Norton & Company, 2008. ISBN: 978-0-393-006228-1.
  - ARMSTRONG, Joe. *Programming Erlang: Software for a Concurrent World*. Pragmatic Bookshelf, 2007. ISBN: 978-1-93435-600-5.
- Artículos y publicaciones:
  - Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia. *Above the Clouds: A Berkeley View of Cloud Computing* (2009).
  - Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg y Ivona Brandic. *Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility* (2008).
  - IEEE Computer Society, Bulletin of the Technical Committee on Data Engineering. *Special Issue on Data Management on Cloud Computing Platforms* (2009).
  - Borja Sotomayor, Rubén Santiago Montero, Ignacio Martín Llorente y Ian Foster. *Capacity Leasing in Cloud Systems using the OpenNebula Engine*.
  - Sun Microsystems. *Take your business to a higher level* (2009).
  - McKinsey&Company. *Clearing the air on cloud computing* (2009).
  - Light Reading's Services Software Insider. *Accelerating Telco services through SaaS/Paas strategies for the SDP in cloud* (2009).
  - Light Reading's Services Software Insider. *Cloud Computing: The fog begins to lift* (2009).
  - Cloud Computing Use Case Discussion Group. *Cloud Computing Use Cases White Paper* (2009).
  - The Economist. *Let it rise, a special report on corporate IT* (2008).
  - InformationWeeks Research&Reports. *Cloud Storage's Top Uses* (2008).

- OpenGridForum Europe. *CloudScape, a Workshop to Explore the Cloud Computing Landscape and its impact on enterprise it* (2009).
- Lamia Youseff, Maria Butrico, Dilma Da Silva. *Toward a Unified Ontology of Cloud Computing* (2008).
- Matthias Brantner, Daniela Florescuy, David Graf, Donald Kossmann y Tim Kraska. *Building a Database on S3*.
- Brian F. Cooper, Raghu Ramakrishnan, Utkarsh Srivastava, Adam Silberstein, Philip Bohannon, HansArno Jacobsen, Nick Puz, Daniel Weaver and Ramana Yerneni. *PNUTS: Yahoo!'s Hosted Data Serving Platform*.
- Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach. *Bigtable: A Distributed Storage System for Structured Data*.
- Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber
- Ian Foster. *What is the grid? A three-point checklist* (2002).
- Bernardo Antonio de la Ossa Pérez. *Erlang: un lenguaje declarativo, concurrente y distribuido para aplicaciones de telecomunicaciones* (2004).

- Páginas web:
  - http://www.nubeblog.com/
  - http://www.ourcloudblog.com/
  - http://asds.dacya.ucm.es
  - http://cloudcomputing.sys-con.com/node/612375/print
  - http://blog.socklabs.com/2008/02/06/dynamically sizing a fragmente.html
  - http://www.virtualizacion.com
  - http://www.linux-kvm.org
  - http://www.xen.org/
  - http://www.google.com/apps/intl/en/business/index.html
  - http://home.live.com/?showunauth=1&mcid=WLWave3Intro
  - http://www.apple.com/mobileme
  - http://www.salesforce.com
  - http://www.sap.com/solutions/sme/businessbydesign/index.epx
  - http://code.google.com/intl/en/appengine
  - http://www.microsoft.com/azure/default.mspx
  - http://www.salesforce.com/platform

- http://www.rollbase.com/home/index.shtml
- http://aws.amazon.com/
- http://www.gogrid.com
- http://www.mosso.com
- http://www.flexiscale.com
- http://www.nirvanix.com
- http://www.opennebula.org
- http://eucalyptus.cs.ucsb.edu
- http://www.abiquo.com/
- http://www.3tera.com/
- http://www.enomaly.com/
- http://www.ubuntu.com/
- http://erlang.org/
- http://www.wikipedia.org/