



Virtualización y alta disponibilidad para entornos de hospedaje y desarrollo de portales web

Virtualization and high-availability environments for hosting and developing Web portals

◆ Jordi Prats, Ricard de la Vega, Joan Cambras

Resumen

Implantación de entornos virtuales mediante el software de código abierto Xen. Se describirán dos entornos en alta disponibilidad: uno para hospedaje de portales web y un segundo entorno para desarrollo y prueba de aplicaciones web.

Palabras clave: entornos virtuales, hospedaje de portales web.

Summary

Deployment of virtual environments using open-source software Xen. We describe two high-availability environments: one for hosting Web portals and a second environment for development and testing of Web applications.

Keywords: virtual environments, hosting Web portals.

◆
Para la creación de entornos virtuales se usa la técnica llamada paravirtualización

1. Introducción

Para la creación de entornos virtuales se usa la técnica llamada paravirtualización. Ésta modifica el sistema para poder ser ejecutado usando un hypervisor. Éste se encarga de presentar una interfaz por software similar al hardware que hay por debajo.

El software de virtualización que se usa es Xen, un software de código abierto desarrollado en sus inicios como un proyecto de investigación en la Universidad de Cambridge. Éste requiere la modificación del sistema operativo para poder ser usado como máquina virtual, aunque manteniendo la compatibilidad binaria de las aplicaciones.

2. Virtualización con alta disponibilidad

Para la implementación de este entorno se han usado nodos HP DL360 G5 con HBAs FC con los que se accede a una SAN. Esto permite arrancar una máquina virtual, y su correspondiente portal web, en cualquiera de los nodos.

Con el objetivo de ofrecer una mayor disponibilidad del sistema virtualizado se ha usado el software RedHat Cluster Suite[1]. Éste permite gestionar tanto el arranque como la migración y parada de las máquinas virtuales. Para ello, comprueba de forma periódica el correcto funcionamiento de las máquinas virtuales y, en caso de detectar un fallo en una de ellas, la vuelve a arrancar, ya sea en el mismo nodo o en cualquier otro que forme el clúster.

◆
Con el objetivo de ofrecer una mayor disponibilidad del sistema virtualizado se ha usado el software RedHat Cluster Suite

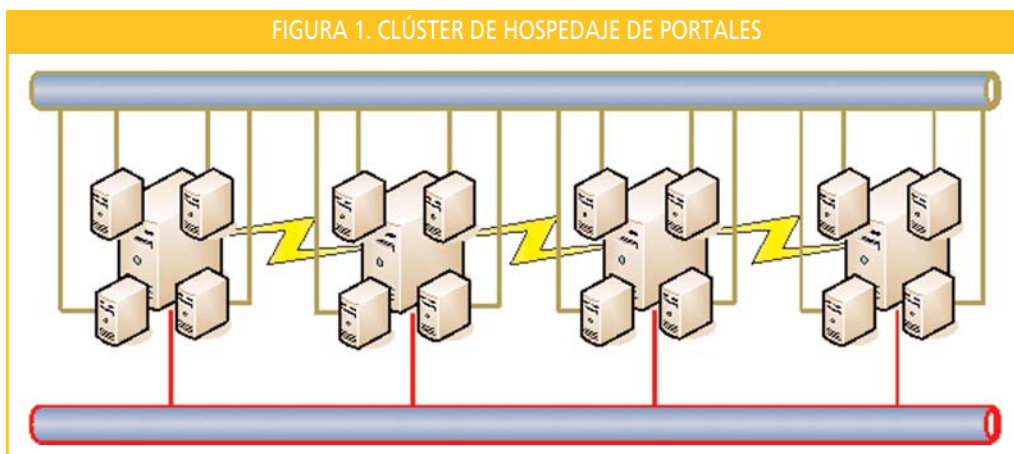
3. Entorno de hospedaje de portales

El entorno de hospedaje de portales se compone de varios nodos (figura 1) para poder repartir las máquinas virtuales entre ellos. Para disponer de alta disponibilidad de las máquinas virtuales, se han sobredimensionado los nodos de tal manera que si falla uno, los otros puedan absorber la carga extra del nodo caído.

Este clúster se ha enfocado a poder ofrecer un entorno dedicado para los usuarios sin dejar de ofrecer una alta disponibilidad. Al implementarlo usando máquinas virtuales, estos usuarios se benefician de un servicio con valor añadido gracias a las características que ofrece la virtualización.

Gracias a la virtualización, se aíslan los distintos portales de manera que no se afectan entre sí. De esta manera, cada uno puede adaptar los parámetros según los requisitos de su aplicación sin que

FIGURA 1. CLÚSTER DE HOSPEDAJE DE PORTALES



éstos provoquen algún problema a los otros entornos. Este aislamiento permite una mejor adaptación a las necesidades de cada cliente (versiones de software, parámetros personalizados, etc.). En cada máquina virtual se pueden instalar las versiones que sean necesarias para la aplicación del cliente e ir actualizándolas al ritmo que el cliente desee. También este aislamiento elimina puertas de entrada a intrusiones, pues cada entorno dispone solamente de los daemons mínimos para ofrecer el servicio.

Las máquinas paravirtualizadas Linux permiten un aumento en caliente del tamaño de la memoria, posibilitando así que los portales que puntualmente necesiten más recursos dispongan de ellos, ofreciendo así una adaptación dinámica y una mejor gestión de los recursos disponibles.

También se puede implementar un entorno de preproducción para desplegar nuevas versiones de las aplicaciones copiando el sistema real. Por ejemplo, desplegar un nuevo campus virtual o probar las actualizaciones del software antes de hacerlo sobre el entorno real para ver las consecuencias.

El uso de snapshots también permite la posibilidad de “empaquetar” una solución y desplegarla en varios entornos: se crea una vez un appliance que realiza una función en concreto que puede desplegarse donde haga falta tantas veces como sea necesario, reduciendo así los costes de puesta en producción del servicio.

Finalmente, en el mantenimiento que debe hacerse regularmente para el buen funcionamiento del sistema que soporta las máquinas virtuales, es posible realizar migraciones en caliente de máquinas virtuales entre nodos, permitiendo realizar el mantenimiento sin cortar el servicio de los portales que mantiene.

A pesar de los claros beneficios de estos entornos no debemos olvidar que también conlleva un cierta penalización en rendimiento debido a la virtualización en sí y un mayor coste de administración, ya que pasa de administrar una sola máquina física a tener que administrar a varias virtuales en una misma máquina física.

De esta manera, se ha pasado de un sistema compartido a un conjunto de entornos dedicados ganando tanto el usuario en disponibilidad, autonomía y flexibilidad, como los técnicos que los administran en flexibilidad y aislamiento entre clientes con requerimientos muy diferentes entre sí.

4. Sistema de desarrollo

Este sistema de desarrollo mejora el proceso de desarrollo de los distintos proyectos. De esta forma, los proyectos pasan por las fases de desarrollo, pruebas y producción. Este esquema permite aislar

Gracias a la virtualización, se aíslan los distintos portales de manera que no se afectan entre sí

Este sistema de desarrollo mejora el proceso de desarrollo de los distintos proyectos



estas fases entre sí, a la vez que gestionar y automatizar al máximo las migraciones para así reducir los periodos de migración y actualización.

Para poder implementar las tres fases del desarrollo de los proyectos se han habilitado en los siguientes entornos independientes con el fin de aislarlos de interacciones entre sí:

Desarrollo: conjunto de máquinas virtuales para los desarrolladores de cada proyecto. En estos casos el desarrollador puede realizar las modificaciones libremente para el desarrollo del proyecto.

Pruebas: réplica del clúster usando máquinas virtuales. Así se reproduce el entorno final que se usará cuando la aplicación esté en producción, pero sin la necesidad de usar varios nodos, sino que uno virtualizado simula el entorno final.

Producción: el propio clúster. Dispone de tres niveles: balanceadores (distribuyen las peticiones), servicios (sirven las peticiones) y datos (almacenan los datos de las aplicaciones).



Para poder evaluar el rendimiento de Xen se han realizado un conjunto de pruebas con un sistema físico y otro virtual

5. Comparativa de rendimiento

Para poder evaluar el rendimiento de Xen se han realizado un conjunto de pruebas con un sistema físico y otro virtual. El físico se trata de un HP DL360G5 con el último kernel disponible de la distribución, con 2 GB de memoria y 4 procesadores Intel Xeon en dos sockets, comparandolo con un sistema virtual con las mismas condiciones.

En la comparación del rendimiento de un sistema virtualizado con uno físico se ha evaluado el rendimiento de una base de datos MySQL en estos sistemas. Aunque el rendimiento de ésta no tenga por qué tener una relación directa, sí que es representativa en el conjunto de sistemas orientados a portales web, por tratarse de una herramienta muy común en éstos.

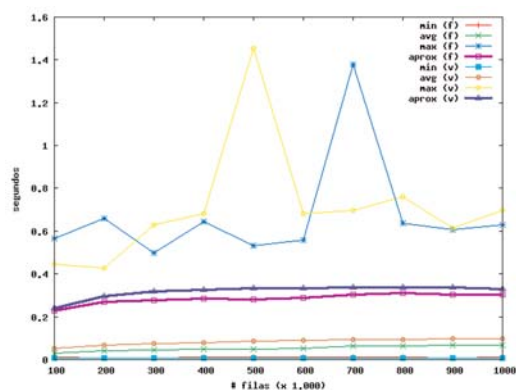
Para ello se ha usado la herramienta de código abierto Sysbench[2]: se trata de una herramienta multi-thread pensada para la realización de pruebas de rendimiento en sistemas pensados para albergar una base de datos. Todas las pruebas se han realizado modificando aleatoriamente los datos para hacer más realista el entorno.



Todas las pruebas se han realizado modificando aleatoriamente los datos para hacer más realista el entorno

En la figura 2 se observa cómo varía el tiempo de respuesta de las consultas de un MySQL cuando aumentamos el número de filas que contiene la base de datos en una máquina física y en una virtual usando 8 clientes simultáneos.

FIGURA 2. TIEMPO DE RESPUESTA DEL MySQL AUMENTANDO EL NUMERO DE FILAS



Los datos que se muestran son las siguientes variables, según los tiempos de respuesta obtenidos pasando una batería de consultas aleatorias a la base de datos: min (tiempo mínimo de respuesta), max (tiempo máximo de respuesta), avg (tiempo medio de respuesta) aprox (tiempo de respuesta esperado): corresponde al mayor número de muestras (95%), por lo que son las que tomamos en consideración como las más representativas del rendimiento del MySQL.

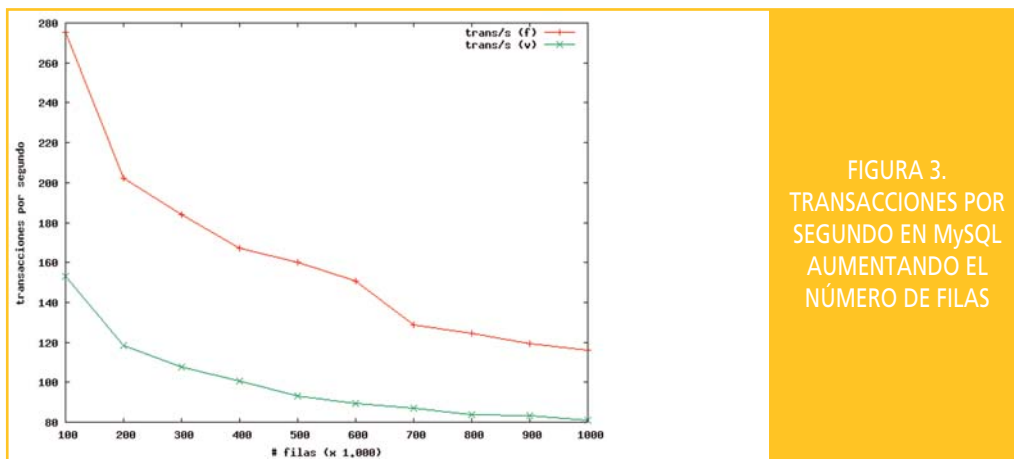


FIGURA 3. TRANSACCIONES POR SEGUNDO EN MySQL AUMENTANDO EL NÚMERO DE FILAS

De acuerdo con los datos, hay una diferencia media de 0,0334 segundos entre la respuesta de la máquina física y la de la máquina virtual, usando como referencia la variable aprox.

En el gráfico de la figura 3 se aprecia el resultado, como en el caso anterior, del aumento del número de filas que contiene la base de datos, pero representando el número de transacciones por segundo que permite la base de datos usando 8 clientes simultáneos. En este caso se aprecia una diferencia media de 63,14 transacciones por segundo de más que puede servir el nodo físico en relación al nodo virtual.

Así, se observa cómo la diferencia de tiempo de respuesta, del orden de la centésima, produce una disminución considerable de las transacciones por segundo.

En el gráfico de la figura 4 se ha utilizado una base de datos con un tamaño de 50.000 registros y se ha medido el tiempo de respuesta variando el número de clientes simultáneos. Igual que en el caso anterior, se han remarcado las líneas que representan el 95% de las muestras. En éstas existe una diferencia media de 0,0564 segundos a favor de la máquina física, excepto en un caso (con 1 solo

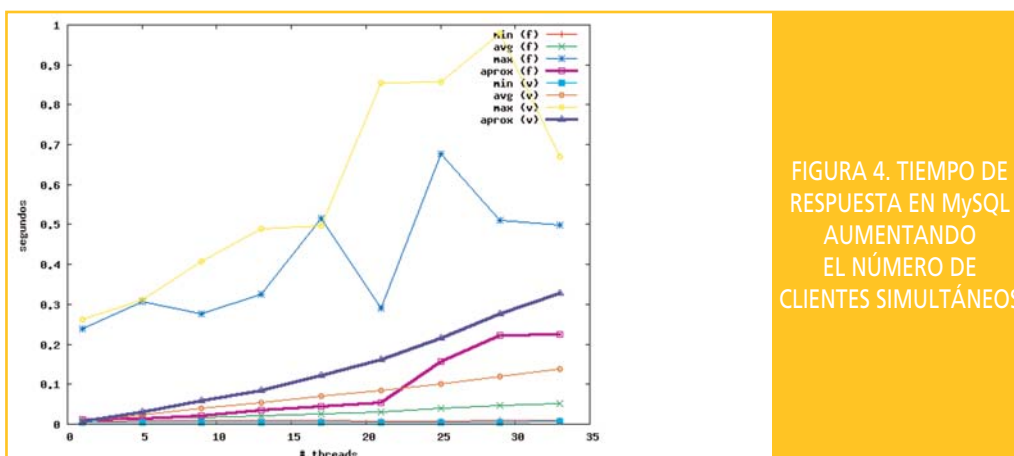


FIGURA 4. TIEMPO DE RESPUESTA EN MySQL AUMENTANDO EL NÚMERO DE CLIENTES SIMULTÁNEOS

La diferencia de tiempo de respuesta, del orden de la centésima, produce una disminución considerable de las transacciones por segundo

En los gráficos se han remarcado las líneas que representan el 95% de las muestras



La virtualización es una técnica que permite una mayor flexibilidad en muchos aspectos, pero que tiene un coste en rendimiento a considerar para extender su uso

Gracias a la virtualización se ha podido adaptar mejor el entorno a los requerimientos de los clientes, así como mejorar el proceso de desarrollo de proyectos

cliente) en el que existe una diferencia de 0,0049 segundos a favor de la virtual. Este caso lo desestimamos por ser poco significativo, ya que con un solo cliente y con tan poca diferencia, se atribuye el resultado a la propia aleatoriedad del sistema de medida.

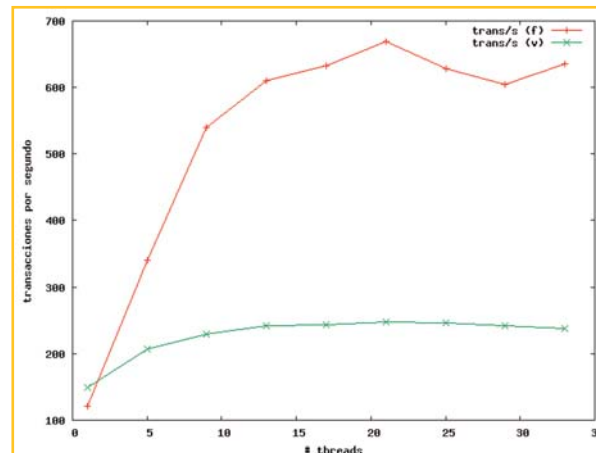


FIGURA 5. TRANSACCIONES POR SEGUNDO EN MySQL AUMENTANDO EL NUMERO DE CLIENTES SIMULTANEOS

Como se aprecia, existe una diferencia del mismo orden, en este caso al comparar la diferencia del tiempo de respuesta incrementado el número de filas, como incrementando el número de clientes simultáneos.

En el caso de las transacciones (figura 5) por segundo, se aprecia también una muestra inicial con la máquina física dando peor rendimiento que una virtual, pero esta muestra puede despreciarse. La media del resto de los valores se sitúa con una diferencia de 276,51 transacciones por segundo a favor de la máquina física, con un máximo de 422,06 transacciones por segundo usando 22 clientes simultáneos.

6. Conclusiones

Según se ha podido ver, la virtualización es una técnica que permite una mayor flexibilidad en muchos aspectos, pero que tiene un coste en rendimiento a considerar y a tener en cuenta para extender su uso. Aún así, es una buena herramienta para servidores poco cargados que quieran consolidarse en un único servidor virtualizado, o para reproducir entornos complejos.

Para conseguir unos niveles altos de disponibilidad, se ha usado una herramienta que actúa automáticamente en caso de fallo para restaurar el servicio. También se ha integrado en el sistema de monitoraje con el fin de reducir el tiempo a partir del momento en que un fallo se produce, se detecta, se corrige y se restablece el servicio. De este modo se persigue la consecución de los niveles de servicio (SLA) adecuados.

En el caso del entorno de hospedaje se ha pasado de un entorno compartido a uno dedicado, aumentando el aislamiento entre proyectos y la flexibilidad para los clientes. Por lo que respecta al sistema de desarrollo, se ha mejorado el entorno de pruebas, ya que se ha podido simular el entorno final donde los proyectos se ponen en producción.

Gracias a la virtualización se ha podido adaptar mejor el entorno a los requerimientos de los clientes, así como mejorar el proceso de desarrollo de proyectos.

Agradecimientos:

El sistema expuesto no hubiera sido posible sin la colaboración de todos los compañeros del Centre de Supercomputació de Catalunya (CESCA). Dar las gracias, especialmente a Jesús Martín, Xavier Torelló y Natalia Torres.

Referencias

- [1] *RedHat Cluster Suite*: <http://sources.redhat.com/cluster>
- [2] *Sysbench*: <http://sysbench.sourceforge.net>.

**Jordi Prats
Ricard de la Vega
Joan Cambras**

Centre de Supercomputació de Catalunya (CESCA)



El sistema expuesto
no hubiera sido
posible sin la
colaboración de
todos los
compañeros del
CESCA