

PRACTICA 7 (FINAL) :

DISEÑO DE UN SISTEMA DE COMPARTICIÓN Y PRESENTACIÓN DE AUDIO ESCALABLE

Última actualización: 3 de diciembre de 2015

Objetivos

- Comprender en mayor profundidad las soluciones de tipo PaaS.
- Implementar soluciones tipo PaaS con un enfoque basado en IaaS.
- Integración de los diversos contenidos de la asignatura.

Descripción

El objetivo de esta práctica es la implementación de un servicio de reproducción de canciones que cumpla el conjunto de requisitos (escalabilidad, fiabilidad, etc.) descritos en este enunciado.

Se desea crear una start-up (CDPSfy) que plantea una red social de intercambio de música en Internet. Para ello nos han encargado el diseño, implementación y despliegue de dicho servicio en la dirección <http://www.cdpsfy.es>.

Como se prevé que un gran número de usuarios usen concurrentemente el servicio, se debe diseñar una arquitectura con una configuración que permita escalar el servicio según crezca el número de usuarios. Por ello se partirá de un sistema con un número determinado de servidores, así como una arquitectura que nos permita añadir servidores (reales o virtuales) según crezca la demanda del servicio.

El servicio a desarrollar debe permitir al usuario subir sus propios ficheros de audio (canciones) y reproducir los ya existentes.

El servicio se divide en dos partes:

- server.cdpsfy.es: es un servidor web que contendrá la lógica de la aplicación así como el modelo de datos de canciones (tracks). Para facilitar la labor al alumno se proporciona un proyecto nodejs-express con una primera versión de la aplicación web implementada:

<https://github.com/aalonsog/CDPSfy>

En el fichero README.md se encuentran las instrucciones de instalación y configuración.

Esta primera versión utiliza un modelo de datos cargado en memoria y sirve los ficheros de audio desde el directorio estático del propio servidor web. El alumno deberá modificar el proyecto para que los ficheros se sirvan y gestionen según la solución diseñada en la práctica. Opcionalmente podrá añadir persistencia al modelo de datos configurando una base de datos.

En la sección “Detalle de la aplicación CDPSfy” se explican las partes de la aplicación que deben modificarse.

- tracks.cdpsfy.es: es una red de servidores, detrás de un balanceador de carga que contendrá una serie de servidores con discos y servidores web. Por simplicidad en la práctica las canciones (tracks) se entregarán usando el protocolo http, en lugar de servidores de streaming. Éstas estarán almacenadas en un sistema ficheros de forma adecuada según los objetivos.

Este servidor debe ser capaz de servir los ficheros de audio almacenados en los discos y de escribir/eliminar nuevos ficheros de audio en dichos discos. Estas acciones se realizarán escuchando peticiones REST (GET, POST, DELETE)

Se pretende que sea un servicio de alta disponibilidad y con una alta tolerancia a fallos. Para ello se utilizará un balanceador de carga frente a, al menos, tres instancias de servidores web.

Con todo esto la arquitectura se puede componer de:

- Una aplicación web con la aplicación node CDPSfy.
- Un balanceador de carga de las peticiones REST a los servidores de ficheros. Debe escuchar en tracks.cdpsfy.es.
- Un conjunto de servidores REST que permitan la lectura, escritura y borrado de canciones con alta disponibilidad, robustez y fiabilidad. Estos servidores estarán detrás del balanceador y gestionarán los discos. Se considerará que disponemos de un centro de datos con 3 servidores con discos que utilizarán el GlusterFS.
- Se añadirá una máquina donde se instalará Nagios para monitorizar toda la infraestructura.

Partes opcionales:

1. Alternativamente (opcionalmente) se puede realizar la practica utilizando Docker, Docker Compose y se puede utilizar Flocker (<https://github.com/clusterhq/flocker>) para la gestión de los volúmenes.

2. También pueden realizarse opcionalmente las mejoras que el alumno considere necesarias en la propia aplicación web node. Algunas sugerencias son:

- Persistencia del modelo de datos configurando y desplegando una base de datos (MySQL, MongoDB...).
- Edición de las carátulas (covers) asociadas a cada canción. Permitir a los usuarios subir una carátula junto al fichero de audio de una canción.
- Gestión de listas de reproducción. Permitir a los usuarios crear listas de reproducción y asociar canciones a ellas.
- Gestión de usuarios. Registro y login. Listas de reproducción privadas.
- Mejoras en el look&feel y UX (user experience). Mejorar el HTML/CSS, los flujos de interacción, añadir iconos, etc.

3. Por último puede realizarse de manera opcional la modificación de la aplicación web node para utilizar una base de datos persistente. En la sección “Detalle de la aplicación CDPSfy” se explican los detalles.

Entrega

Los alumnos deberán entregar un documento donde se describa la arquitectura propuesta, así como algunas ideas sobre el despliegue del mismo en el laboratorio. En particular, el documento incluirá:

- Pasos seguidos en la implementación de la solución
- Aplicación CDPSfy modificada.
- Diseño de la solución de alta disponibilidad partiendo de las ideas aprendidas dentro de la asignatura. Se debe representar mediante al menos una figura la arquitectura de las máquinas y las funciones que realizan. Además, se incluirá una breve descripción de los paquetes y servicios que se ejecutarán en casa máquina.
- Capturas de pantalla con las configuraciones que hay que realizar para implementar la solución.
- Se indicará como cambiaría el despliegue en caso de desplegarlo en OpenStack y en Amazon AWS.

Esta documentación se deberá entregar antes de las 23:59 del lunes 11 de enero de 2016.

La evaluación de la práctica se realizará a partir del jueves 21 de enero. Se publicarán oportunamente los turnos disponibles. Se evaluará el correcto funcionamiento del servicio, la calidad de la solución adoptada en relación a los requisitos definidos en este documento y el conocimiento de los alumnos de las técnicas y herramientas usadas en el desarrollo de este trabajo.

Detalle de la aplicación CDPSfy

La versión de la aplicación CDPSfy que se proporciona está configurada con un modelo de datos en memoria y sirve los ficheros de audio desde el directorio estático de la aplicación. Además las funcionalidades de subir y borrar ficheros no están implementadas en su totalidad.

El alumno deberá modificar el controlador de tracks (CDPSfy/controllers/track_controller.js) para que la gestión de ficheros de audio se realice a través de la solución diseñada para *tracks.cdpsfy.es*. En el propio controlador pueden encontrarse algunas instrucciones. Básicamente se trata de modificar los métodos de creación y borrado de *tracks*:

- *create*: en la versión proporcionada este método simplemente añade al modelo de datos local los metadatos de la nueva canción. Debe modificarse el método para:
 - o Escribir el fichero en *tracks.cdpsfy.es* (según la solución implementada por el alumno).
 - o Introducir en el modelo de datos la verdadera url generada para la canción en *tracks.cdpsfy.es*.
- *destroy*: en la versión proporcionada este método simplemente elimina la entrada correspondiente a la canción del modelo de datos local. Debe modificarse para eliminar el fichero de *tracks.cdpsfy.es*

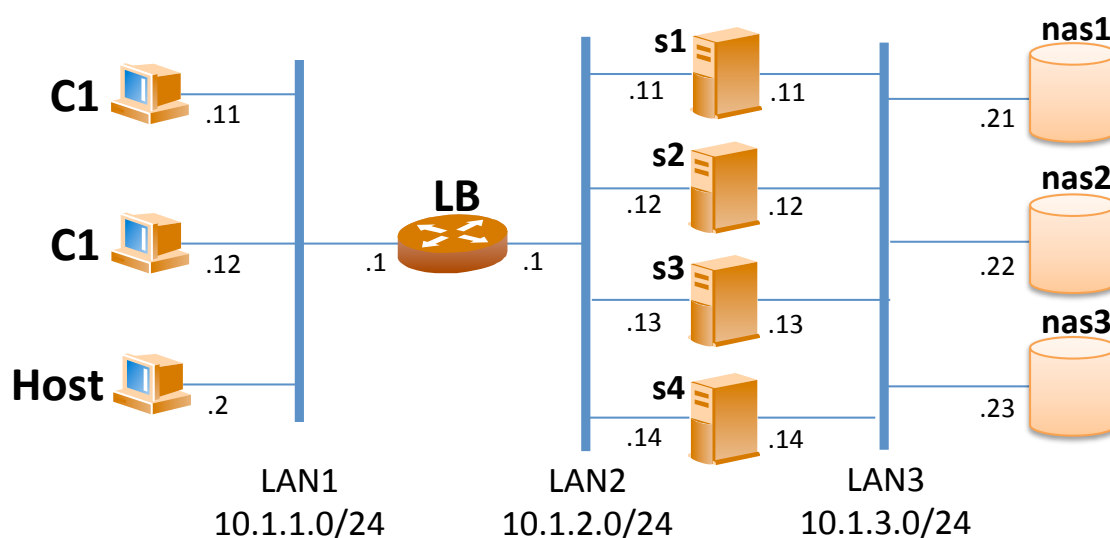
Con esta solución los metadatos de las canciones se guardan en memoria (según el modelo definido en *CDPSfy/models/track.js*) por lo que al reiniciar el servidor node perderemos las actualizaciones. Se propone como mejora opcional implementar un modelo de datos persistente utilizando cualquier base de datos conectada al servidor *server.cdpsfy.es*. Algunas sugerencias son:

- MySQL: puede utilizarse el driver *sequelize* (<http://docs.sequelizejs.com>) con el que el alumno estará familiarizado gracias a la asignatura CORE.
- MongoDB: puede utilizarse el driver *Mongoose* (<http://mongoosejs.com>)

Escenario de la práctica

La práctica se realizará mediante el uso de escenarios virtuales formados por máquinas virtuales interconectadas siguiendo la topología típica usada en aplicaciones escalables y redundantes. Para facilitar la realización de la práctica se va a utilizar la herramienta VNX de gestión de escenarios virtuales (vnx.dit.upm.es), que permite la automatización del arranque y configuración de escenarios. En particular, se ha preparado el escenario virtual de la figura, compuesto por dos máquinas cliente, un balanceador de carga, tres servidores www y tres servidores de disco, todos ellos implementados como máquinas virtuales ligeras (basadas en LXC).

La práctica podrá realizarse en el laboratorio o en un ordenador propio. En el laboratorio, los escenarios pueden arrancarse directamente desde los puestos de prácticas. Para facilitar la realización en ordenadores propios se ha preparado una máquina virtual para VirtualBox con Ubuntu 14.04 y que tiene instaladas todas las herramientas necesarias para arrancar el escenario virtual.



Para arrancar el escenario de la práctica siga las instrucciones siguientes:

1. A) Si utiliza ordenador propio:

- Descargue e instale VirtualBox desde <http://www.virtualbox.org>. Debe instalar además el “VM VirtualBox Extension Pack” disponible también en la página de descargas. Si ya tiene instalado VirtualBox, se recomienda actualizarlo a la última versión.
- Descargue la máquina virtual a su ordenador desde el enlace disponible en el moodle de CDPS.

- Arranque VirtualBox y acceda al menú "Archivo->Importar servicio virtualizado", seleccione el fichero de la máquina virtual y a continuación dele al botón "Importar".
- Una vez finalizada la importación, seleccione la máquina virtual y haga doble click sobre ella o pulse el botón "Iniciar" para arrancarla.

B) Si utiliza el laboratorio, simplemente acceda a su cuenta y arranque el escenario según se describe a continuación.

2. Para gestionar el escenario virtual de la práctica, utilice los comandos siguientes:

- a. Descargue el escenario y descomprima el fichero:

```
wget http://idefix.dit.upm.es/download/cdps/p7/p7.tgz
tar xfvz p7.tgz
cd p7
```

- b. Para arrancarlo:

```
sudo vnx -f p7.xml -v -create
```

- c. Para acceder a las máquinas virtuales del escenario puede hacerlo a través de las consolas que aparecen al arrancarlo o también mediante SSH. Por ejemplo, para acceder a s1:

```
ssh root@s1
```

IMPORTANTE: Utilice la cuenta 'root' con clave 'xxxx' para acceder a las máquinas virtuales.

- d. Para pararlo conservando los cambios realizados en las MVs:

```
sudo vnx -f p7.xml -v --shutdown
```

Tenga en cuenta lo siguiente: si realiza la práctica en la máquina virtual, los cambios se guardan en la misma, por lo que se conservarán en sucesivos arranques. En cambio, si realiza la práctica en el laboratorio, los cambios solo se conservarán mientras permanezca en el mismo puesto. Si cambia de puesto o utiliza el mismo puesto en días sucesivos, los cambios se perderán.

- e. Para pararlo borrando los cambios realizados en las MVs:

```
sudo vnx -f p7.xml -v --destroy
```

- f. Para mostrar el mapa del escenario:

```
sudo vnx -f p7.xml -v --show-map
```

3. Tenga en cuenta que todas las máquinas virtuales del escenario utilizan el mismo sistema de ficheros (root filesystem), que ya tiene instalados los principales paquetes necesarios. Si necesita instalar algún paquete nuevo en la imagen de las máquinas virtuales, puede instalarlo de la forma habitual (apt-get install ...) ya

que las máquinas virtuales tienen conexión a Internet. De todas formas, tenga en cuenta que, tal como se ha descrito, los cambios realizados en las máquinas virtuales pueden perderse al parar el escenario.

En el caso de la máquina virtual, es posible instalar nuevos paquetes de forma permanente mediante el procedimiento siguiente:

- Parar el escenario de la práctica con la opción `--destroy` (IMPORTANTE: se perderán todos los cambios realizados en las MVs)

```
sudo vnx -f p7.xml -v --destroy
```

- Arranque una máquina virtual que utilice en modo directo y con conectividad a Internet la imagen de las MVs:

```
sudo lxc-start -n cdps -f /usr/share/vnx/filesystems/rootfs_lxc/config
```

- Una vez arrancada, active la red con:

```
sudo dhclient eth0
```

- A continuación instale todo lo que considere necesario. Una vez terminada la instalación, pare la máquina virtual con `"halt -p"`. Todos los cambios que haya realizado quedarán permanentes en la imagen y, por tanto, serán visibles desde todas las máquinas virtuales.

4. Es posible ejecutar comandos en las máquinas virtuales desde el host mediante el comando `"lxc-attach"`. Por ejemplo:

```
sudo lxc-attach -n s1 -- /etc/init.d/apache2 start
```

Asimismo, es posible copiar ficheros directamente desde el host a una máquina virtual, dado que los sistemas de ficheros de las MVs ligeras se ven como directorios en el host. Por ejemplo, para copiar un fichero `/tmp/fichero` desde el host al directorio `/tmp` de la máquina virtual `www1`:

```
cp /tmp/fichero /var/lib/lxc/www1/rootfs/tmp
```

5. Para mejorar el funcionamiento de la maquina virtual VirtualBox, se recomienda (siempre en función de los recursos que posea la máquina en la que se ejecute) aumentarle la memoria (2Gb o más si es posible) y el número de CPUs (más de 2 si es posible).
6. Es posible modificar el escenario de la práctica añadiendo, por ejemplo, más servidores si lo cree conveniente. Para ello, edite el fichero de definición del escenario (`p7.xml`) y replique la definición de las máquinas virtuales (bloques definidos por la etiqueta `<vm>`).

Configuración de GlusterFS

Se incluyen a continuación una serie de recomendaciones sobre la configuración del sistema de ficheros glusterfs en los servidores nasX y sobre como montar ese sistema de ficheros desde los servidores (sX).

1. Configuración de servidores de disco (nas)

- Para simplificar, los servidores de disco no tienen múltiples discos y, por tanto, no es necesario configurar ningún tipo de RAID en ellos. Se utilizará un directorio del sistema de ficheros de la máquina virtual (/nas por ejemplo) que se exportará y sincronizará con el resto de servidores.
- Cada servidor nas, tiene que tener configurado un identificador (uuid) distinto en el fichero /etc/glusterd/glusterd.info. Dado que todas los nas parten de la misma imagen, tienen configurado el mismo uuid. Por ello, es necesario generar dos nuevos uuid (usando el comando *uuidgen*) y configurarlos en los nas2 y nas3. Y rearrancar posteriormente el servicio para que los cambios tengan efecto:

```
service glusterfs-server restart
```

- Para crear el cluster de servidores de disco replicados se pueden usar los comandos vistos en la practica 3 y realizarlo enteramente desde el nas1. Añadimos los servidores al cluster mediante:

```
gluster peer probe 10.1.3.XX
```

Vemos el estado de los servidores con:

```
gluster peer status
```

Creamos un volumen con tres servidores que replican la información con:

```
gluster volume create nas ... (ver resto de parámetros en manual)
```

Arrancamos el volumen con:

```
gluster volume start nas
```

Vemos el estado de los volúmenes creados con:

```
gluster volume info
```

2. Configuración del montaje desde servidores web (www)

- Para acceder al sistema de ficheros exportado por los nasX desde los servidores de web se puede utilizar el comando mount. Por ejemplo:

```
mkdir /mnt/nas  
mount -t glusterfs 10.1.3.21:/nas /mnt/nas
```


- Una vez ejecutado ese comando, desde el directorio /mnt/nas del servidor web se verá el contenido del directorio exportado por los nas. Si se copia un fichero a ese directorio, se puede comprobar desde la consola de cada nas que fichero copiado se replica a todos los servidores.
- **Importante:** la sincronización de ficheros entre los distintos nas solo funciona si los ficheros se copian/modifican/borran desde los servidores que montan el directorio compartido. Si modificáis los ficheros directamente desde los servidores nasX no funciona.
- El montaje de los nas mediante el comando mount no es la forma más adecuada, ya que hay que especificar en el comando el servidor con el que queremos contactar (nas1, nas2, nas3). Si ese servidor nas se cae, el servidor www dejará de poder acceder al disco. Para evitar este problema se sugiere utilizar uno de los dos métodos propuestos en:

<http://www.jamescoyle.net/how-to/439-mount-a-glusterfs-volume>