Clientes ligeros para máquinas virtuales sobre Spice





Autor: Jose Enrique Ruiz Navarro

Jose Enrique Ruiz Navarro

Table of Contents

Clientes ligeros para máquinas virtuales sobre Spice	1
Introducción	3
¿Qué es un cliente ligero?	4
Virtualización de escritorios	5
¿Qué es?	5
Comparación de las diferentes tecnologías	6
Planteamiento del problema	6
Idea principal	7
¿Qué es Spice?	8
Historia de Spice	9
Componentes de Spice	9
Cómo funciona	10
Escenario	12
Servidor Spice	13
Instalación de Spice	13
Preparación de máquinas	13
Cliente Spice	19
PXE	23
¿Qué es?	23
Funcionamiento	23
Construcción de la distribución	24
Ubuntu-builder	24
Instalación de los servicios en el servidor	
DHCP	
Tftp-hpa	
Construcción de imagen para arranque PXE	32
NFS	35
Problemas encontrados	
Spice con Openstack	
Raspberry pi como cliente ligero	
Redirección de USB	
Conclusión	
Bibliografía	41

Introducción

La complejidad del software y los requerimientos de computación son cada vez mayores. Consecuentemente, las características del hardware a utilizar crecen en la misma medida. Incluso los ordenadores, tanto de escritorio como portátiles, que se usan actualmente a nivel doméstico, están dotados de una potencia y capacidad habitualmente mucho mayor que la necesaria para el uso al que están sometidos. Esto da como resultado el desperdicio innecesario de recursos, y a su vez, un aumento del costo de material informático para la empresa. Estas máquinas, como todas también se deterioran y a la larga se tienen que sustituir por otras.

Este planteamiento, basado en la adquisición de hardware desproporcionado, y su reemplazo frecuente, conlleva un coste, esfuerzo y tiempo que pueden ser minorados en gran medida.

Una solución para reducir costos y recursos, es proveer al usuario con un escritorio remoto. Los usuarios ya no utilizarán un ordenador caro e infrautilizado, sino que será un servidor central el que ofrezca el entorno de trabajo. Aún más, la adopción de esta filosofía constituye, en gran número de ocasiones, una oportunidad de reutilizar o mantener en activo elementos informáticos que, de otra forma, quedarían obsoletos o fuera de uso.

El modelo de implantación que se propone mejorará en muchos casos, y siempre sometido a un estudio previo de las necesidades de computación del entorno de trabajo, tanto los costes de adquisición y reemplazo de material informático, como los de mantenimiento y configuración, consecuencia directa de la centralización del sistema operativo. Dicho informe deberá reunir también las limitaciones de uso de esta estructura, que implica la dependencia absoluta de funcionamiento de una sola máquina central, así como la pobreza de aplicaciones

3

de uso multimedia, como audio o vídeo de alta calidad.

Adicionalmente, y para maximizar estos resultados positivos, todas las arquitecturas a considerar estarán basadas en software libre, siempre más seguro y estable que sus contrapartidas propietarias. Dichas arquitecturas serán: Ltsp, Openthinclient, Spice y VNC.

¿Qué es un cliente ligero?

La definición de cliente ligero engloba tanto un software como un equipo físico que utiliza los recursos de otro equipo, (el servidor de escritorio), para hacer la mayor parte de su tarea. En un cliente ligero se tiene que ejecutar un software, el cual se conectará a un servidor, que es el que estará realizando todo el trabajo, todas las aplicaciones que se muestran en la pantalla del cliente ligero. Por lo tanto un cliente ligero es un dispositivo que precisa de pocos recursos físicos, en el que se mostrará un escritorio que se está ejecutando en un servidor.



Virtualización de escritorios

¿Qué es?

En términos generales, la virtualización de escritorio describe el proceso de separación entre el escritorio, que es donde se encuentran los programas que los usuarios ejecutan, de la máquina física (que sería el cliente ligero). El escritorio "virtualizado" se almacena de forma remota en un servidor central en lugar de en el disco duro del ordenador personal. Esto significa que cuando los usuarios trabajan en su escritorio desde su puesto de trabajo, todos sus programas, aplicaciones, procesos y datos se almacenan y ejecutan en el servidor, permitiendo a los usuarios acceder remotamente a sus escritorios desde cualquier dispositivo capaz de conectarse remotamente, tales como un portátil, PC, smartphone o Tablet.

Comparación de las diferentes tecnologías

	Open ThinClient	LTSP	VNC	Spice
Código abierto	Si	Si	Si	Si
Desarrollado en	Java	Shell script	C++/Java	C/C++
Soporte USB en cliente	Si	Si	No	Si
Soporte audio en cliente	Si	Si	No	Si
Exportación de la interfaz gráfica de usuario	SSH -X	Xdcmp	RFB (Remote Frame- Buffer)	Protocolo Spice
Software en la parte del cliente	No hace falta	No hace falta	Cliente VNC	Cliente Spice
Implementación en cliente ligero	Si	Si	No	¿No?
Solución disponible para navegador	No	No	Si	Si

Planteamiento del problema

En las soluciones más comunes, es el propio servidor quien genera una sesión virtual para cada usuario y la muestra en el cliente que lo solicita. Pero claro, sin conocer el uso que el usuario va a darle a esa sesión previamente. Por ejemplo, si se parte de la suposición que un usuario puede lanzar una aplicación que consuma bastante recursos del servidor, obviamente los otros usuarios que tengan abierta una sesión, notarán como el sistema se ralentiza. Otro posible caso sería contemplar que hubiese un mal uso en una sesión y un virus infecte al equipo, este incidente afectaría a todos los usuarios.

Una alternativa sería beneficiarse del aislamiento que ofrecen las máquinas virtuales, asignándolas directamente a cada usuario. De este modo, si pasara alguna situación como las de los ejemplos anteriores, perjudicaría sólo a su máquina virtual.

Sabiendo que solamente se desea que una máquina virtual corresponda a un usuario, habría que descartar Ltsp y Openthinclient. Por lo tanto, restarían los otros dos protocolos que hemos estudiado; VNC y Spice.

Hay que considerar que en los entornos de trabajo se hace cada vez un mayor uso de contenidos multimedia, lo que nos llevaría a descartar VNC, que carece de características como transmisión de audio y una aceptable velocidad de transmisión de vídeo.

Finalmente, sólo quedaría Spice, una opción que es relativamente nueva. Como se ha podido observar en la comparativa de las tecnologías, Spice no tiene solución en clientes ligeros, puesto que requiere un software (cliente Spice) para conectarse a la máquina remota.

Y así se concreta la idea de este proyecto: hacer posible que a través de Spice, un cliente ligero se conecte a una máquina virtual.

Idea principal

Sabiendo que Spice no dispone de una solución para clientes ligeros, buscaremos la forma de que a través del protocolo Spice, un cliente ligero se conecte al servidor sin que tenga sistema operativo alguno.

Por lo tanto se contará con un servidor que albergará distintas máquinas virtuales. Estas máquinas se visualizarán remotamente mediante Spice. Este mismo servidor ofrecerá una distribución de Linux, preparada previamente, por PXE con la activación del cliente Spice al inicio. Así pues, cuando se arranque un dispositivo por PXE, el servidor le ofrecerá un menú con las máquinas virtuales

disponibles y al elegir una de ellas, accederá directamente a ella.

En resumen, un dispositivo sin sistema operativo se conectará a la red, este mismo a través de PXE obtendrá un sistema operativo preparado para que arranque el cliente Spice, y finalmente inicie la conexión a una máquina determinada.



¿Qué es Spice?

Según Red Hat, Spice(simple protocol for independent computing enviroment) es un protocolo de escritorio remoto de alto rendimiento, dinámico y adaptable capaz de ofrecer una experiencia de usuario final idéntica a la obtenida con un PC de escritorio físico.

Historia de Spice

Spice fue originalmente desarrollado por la empresa Qumranet, quien en su época de independencia corporativa se dedicaba a desarrollar entornos de virtualización de escritorios basados en el uso de KVM (Kernel Virtual Machine). Suena bastante lógico que esta empresa haya trabajado sobre este protocolo si tenemos en cuenta que fueron ellos mismos los que crearon y mantuvieron nada menos que el mismísimo hipervisor KVM, que hoy en día podemos ver en sistemas operativos GNU/Linux y hasta en OpenIndiana gracias a los aportes y el desarrollo de la empresa Joyent.

Este producto, Spice, surgió como un subconjunto de su idea original, Solid ICE, que en su momento parecía estar posicionándose como el producto de virtualización de escritorios por excelencia, dado que entregaba sus interfaces mediante el uso de una aplicación web o un pequeño cliente ligero. Como todos nos podemos imaginar, ICE es parte del nombre Spice, justamente por ser el acrónimo de "Independent Computing Environment" (entorno de computación independiente).

Y como muchas veces pasa en el mundo corporativo, esta pequeña empresa israelí que armaba soluciones de código cerrado para GNU/Linux fue comprada en el año 2008 por Red Hat, quien un tiempo después integró su cartera de productos con los que Qumranet poseía. Por supuesto, abrió su código a las comunidades, permitiéndoles alimentarlo y acrecentarlo para llegar a ser lo que es ahora.

Componentes de Spice

9

La arquitectura Spice está compuesta por estos componentes:

- Servidor Spice: Este componente es el que envía la imagen al cliente.
- Cliente Spice: Este otro estará en el lado del cliente. Es el encargado de traer del servidor el escritorio remoto y trasladarlo donde se esté ejecutando.
- Spice Device: Determina el hardware virtual a emular en cada máquina, de modo que el cliente adopte la resolución de pantalla, tipo de teclado, ratón, y en su caso salida de audio y micrófono configurada en la máquina virtual.
- Spice Driver: Es el conjunto de controladores de hardware para los dispositivos determinados por Spice Device.
- Spice Agent: Virtualiza las interfaces de la propia máquina virtual.
- Protocolo Spice: Por último, el cliente usa el protocolo para conectarse al servidor y comunicar todos los datos del escritorio (vídeo, audio, dispositivos de entrada como ratón y teclado...).

Cómo funciona

El cliente se conecta al componente servidor Spice, que estará en el equipo anfitrión, a través del protocolo Spice. Una vez la petición llega al servidor, este pregunta a Spice Device, que le comunica qué dispositivos tiene disponible para ofrecer. Spice device se conecta con Spice driver, que aporta los drivers de hardware. Por último Spice driver, recibirá de spice agent los datos de la máquina que ofrecerá al cliente.

Jose Enrique Ruiz Navarro



Cuando el cliente se conecta al servidor, lo hace a través de varios canales. Cada canal está dedicado exclusivamente a un tipo de dato, así el ratón y teclado irán por uno diferente al del audio.

Los canales más importantes son:

- Main: Este canal está implementado por RedClient.
- DisplayChannel: Es el responsable de gestionar los comandos gráficos, así como las imágenes y los flujos de video.
- InputsChannel: Es quien gestiona la entrada por teclado y mouse.
- CursorChannel: Es el que maneja la posición y visibilidad del cursor.
- PlaybackChannel: Para nuestra grata alegría, este canal maneja la forma en la cual el servidor envía audio y video al cliente de forma en que permita ver, por ejemplo, vídeos remotamente sin prácticamente ningún tipo de retardo.
- RecordChannel: Es el encargado de capturar audio y vídeo del lado del cliente, y enviarlo al servidor.

Jose Enrique Ruiz Navarro



Escenario

El escenario estará formado por el servidor, que tendrá varias máquinas virtuales. Este servidor estará basado en Ubuntu 12.10, puesto que en la versión 12.04, la instalación de los paquetes de Spice genera errores. Además, en este servidor, estarán instalados los servicios TFTP , DHCP y NFS , que son los que permitirán que los clientes arranquen vía PXE la distribución que se ha preparado.

Como equipo cliente servirá cualquier dispositivo (de escritorio, portátil...) que arranque mediante PXE.

Servidor Spice

Instalación de Spice

Instalaremos los paquetes de virtualización:

apt-get install kvm libvirt-bin virt-manager virt-viewer

A continuación instalaremos los paquetes necesarios para que las máquinas se puedan visualizar a través de Spice:

apt-get install qemu-kvm-spice python-spice-client-gtk
spice-client-gtk libspice-server1 libspice-client-gtk-2.0-1

Libspice-server se instalará por dependencia al igual que libspice-client-gtk-2.0-1. Spice-client-gtk se instala para que cuando se creen las máquinas virtuales, se pueda visualizar en el mismo.

Preparación de máquinas

Hay que preparar la máquina virtual para que se pueda visualizar por Spice.

Desde línea de comandos se podrá instalar introduciendo la siguiente sentencia:

13

```
Proyecto integrado
```

Jose Enrique Ruiz Navarro

virt-install --name prueba --hvm --vcpus=1 --ram 1024 --disk
path=/home/quique/prueba.img,size=10 --network network=red
--os-type linux --noautoconsole --graphics
spice,port=5900,listen=0.0.0.0
--cdrom=/home/quique/Escritorio/ubuntu-12.04.2-desktop-i386.i
so

Como se puede observar, en graphics le asignamos el modo de visualización, que será VNC o Spice, el puerto en el que escuchará el servidor (no se pueden configurar dos máquinas en el mismo puerto porque dará error al intentar instalar la segunda) y la red que permitirá el acceso. La anterior sentencia nos abrirá una pantalla de instalación del sistema de la máquina virtual.

También se puede instalar a través de virt-manager, sólo hay que cambiar el modo de visualización. Antes de instalar la máquina, se personalizarán los componentes.

Ready to begin installation of **prueba** SO: Ubuntu 12.04 LTS (Precise Pangolin) Instalar: CDROM/ISO local Memoria: 1024 MB CPUs: 1 Almacenamiento: 8.0 GB /var/lib/libvirt/images/prueba-1.img Solid Personalizar configuración antes de instalar

El vídeo será de modelo QXL.

🎻 Iniciar la instalación	🔀 Cancelar
 Overview Processor Memory Boot Options Disk 1 NIC :e0:89:3a Entrada Sound: default Console Vídeo 	<pre>Video Modelo: qxl RAM: - Cabezas: -</pre>

El monitor por defecto es VNC, así que se elimina el hardware y seleccionamos el tipo servidor Spice. Aquí se elegirá el puerto, la contraseña y donde estará escuchando el servidor.

Jose Enrique Ruiz Navarro

	Storage Network	Gráficos
0	Input	Indique cómo le gustaría visualizar su pantalla virtual.
2	Graphics	Tines Consider Solice
1	Sound	Tipo: Servidor Spice
	Serial	Sugerencia: VNC o el servidor Spice es muy recomendable ya sue possible la visualización vistual para ser insurtado dentre de
	Parallel	esta aplicación. También se puede usar para permitir el acceso a
	Channel	la pantalla virtual desde un sistema remoto.
10	USB Host Device	Dirección: 👿 Escuchando en todas las interfaces de red públicas
10	PCI Host Device	
-	Video	Puerto ILS S931 - Asignados automaticamente
	Watchdog	Contraseña:
	Filesystem	
	Smartcard	Mapa de teclado: 🗹 Igual que el anfitrión 🛛 Otro:
Ð	USB Redirection	
		Cancelar Finalizar

Se puede observar en el xml de la máquina virtual (/etc/libvirt/qemu/máquina.xml) lo que ha sido modificado:

```
</channel>
```

Jose Enrique Ruiz Navarro

Como se ve en el apartado de funcionamiento de Spice, la máquina virtual en sí tendrá el agente spice. Esto en una máquina Linux ya viene instalado debido a que qemu se lo incluye, pero en Windows no se instala, por lo que el escritorio de esa máquina no virtualiza los dispositivos.

Por lo tanto, en las máquinas con SO Windows, hay que instalar spice agent.

El driver lo podemos encontrar en :

http://www.spice-space.org/download/binaries/spice-guest-tools-0.1.exe

		-
Ġ Atrás 🝷 💿 🝸 🛃 💋 🔎 Búsqueda 🤺 Favoritos 🤣 😥 è 😹 🔜 🕉		
Dirección 🕘 http://www.linuxito.com.ar/gnu-linux/nivel-alto/58-utilizar-spice-en-maquinas-virtuales-kvm	💌 🄁 Ir	Vínculos
Clashe-taconine		2
Descarga de archivo - Advertencia de seguridad 🛛 🔀		
¿Desea ejecutar o guardar este archivo?		
Tipo: Aplicación, 17.1 MB		
Be: www.spice-space.org		
	.	4.4
descargar el driver QXL para V	a mistalacioni se	ueoe
Desde la máquina virtual desca		
Los archivos procedentes de Internet pueden ser útiles, pero este tipo de archivo puede dañar potencialmente su equipo. Si no contre en el origen, no ejecute ni guarde este software. ¿Cuál es el riesgo?		
Life Lot gevr layerses Look Lop	1	
🛞 tazz 🕗 🕤 🎓 😥 🔎 Sada i 🎼 🕬 klas 🔤 🕬		

Una vez descargado, lo ejecutamos:

Se mostrará un asistente de instalación. Si aparece la imagen de a continuación se deberá pulsar *continuar*.

Jose Enrique Ruiz Navarro

Proyecto integrado



En el administrador de dispositivos se puede observar cómo el driver QXL está instalado.

Se reinicia la máquina virtual y tras su inicio se comprobará que spice agent está en funcionamiento.

Cliente Spice

Para acceder a cualquier máquina virtual, es necesario un software que actúe como cliente. Existe tanto para Linux como para Windows. Spice es un desarrollo de Red Hat, así que puede ser que no esté incluido en los repositorios de todas las distribuciones. Para Windows está disponible en formato ejecutable.

Para instalar el cliente en Ubuntu:

apt-get install spice-client

Para conectarse a una máquina virtual:

Jose Enrique Ruiz Navarro

spicec -h 192.168.1.13 -p 5900

-h:lp del servidor

-p: Puerto en el que está escuchando el servidor para esa máquina virtual.

Se puede instalar también la GUI del cliente, que facilitará el manejo al usuario con menos conocimientos en línea de comandos.

Jose Enrique Ruiz Navarro

apt-get install spice-client-gtk

Una vez instalado, ejecutaremos en la línea de comandos, Spicy

	Connect to SPICE
Hostname	localhost
Port	
TLS Port	
Recent connections:	
localhost?port=5900;	
192.168.10.41?port=5900	D;
192.168.10.41?port=5903	1;
192.168.10.41?port=5902	2;
	Cancelar Conectar

En clientes para Windows o para alguna distribución basada en Red Hat el cliente es diferente.

En la imagen inferior podemos observar como hay que indicar el protocolo, la ip y finalmente el puerto:

Jose Enrique Ruiz Navarro

	Connection details	×
URL:	spice://192.168.10.41:5901	
Recent connections		
spice://192.168.10.4	41:5901	
	Cancel Connect	t

Una vez que se ha conectado el cliente, en el servidor se puede observar en el log de la máquina virtual (/var/log/libvirt/qemu/máquina.log), cómo inicia la conexión:

nain_channel_link: add main channel client
nain_channel_handle_parsed: net test: latency 4.488000 ms, bitrate 52481869 bps (50.050611 Mbps)
inputs_connect: inputs channel client create
red dispatcher set cursor peer:

Se puede contemplar en la parte del servidor cómo establece varios canales:

root@pr	oyecto:/h	ome/quique# netstat -tu	pan grep spice		
tcp	Θ	0 0.0.0.0:5900	0.0.0.0:*	ESCUCHAR	2608/kvm-spice
tcp	Θ	0 0.0.0.0:5901	0.0.0:*	ESCUCHAR	2640/kvm-spice
tcp	Θ	0 192.168.1.2:5900	192.168.1.40:34968	ESTABLECIDO	2608/kvm-spice
tcp	Θ	0 192.168.1.2:5900	192.168.1.40:34969	ESTABLECIDO	2608/kvm- spice
tcp	Θ	0 192.168.1.2:5900	192.168.1.40:34972	ESTABLECIDO	2608/kvm-spice
tcp	Θ	0 192.168.1.2:5900	192.168.1.40:34970	ESTABLECIDO	2608/kvm-spice
tcp	Θ	0 192.168.1.2:5900	192.168.1.40:34971	ESTABLECIDO	2608/kvm-spice
tcp	Θ	0 192.168.1.2:5900	192.168.1.40:34973	ESTABLECIDO	2608/kvm-spice

Jose Enrique Ruiz Navarro

Proyecto integrado

PXE

¿Qué es?

Preboot eXecution Environment o entorno de ejecución de prearranque es, como indica su denominación, un entorno para arrancar/instalar un sistema operativo por red.

Funcionamiento

El cliente solicita dirección de red al servidor dhcp, éste se la facilita pero junto

con la ruta de un fichero para inicializar el arranque por red. Procesa la información obtenida por dhcp y le solicita al servidor el fichero indicado en la respuesta dhcp. El servicio TFTP le envía el fichero al cliente donde posteriormente se procesará. Una vez procesado, descarga el fichero que arrancará por defecto, en este caso será el menú donde se podrá elegir la distribución que quiera el usuario. Finalmente, tras la elección, descargará el kernel, el initrd y montará el sistema operativo que se haya elegido a través de NFS.

Construcción de la distribución

Se ha decidido usar una distribución Ubuntu en modo "probar Ubuntu", de este modo, cuando el cliente ligero requiera por red un sistema operativo, le será entregado Ubuntu en modo prueba. Pero el usuario tendría que instalar spice-client cada vez que se quisiera conectar, por lo que esta distribución, tal como se facilita, no es muy conveniente. Además de este problema, será preciso eliminar software innecesario y reducir el tamaño de la distribución (y con ello el tiempo de transmisión de servidor a cliente y el impacto sobre la red), por lo que se construirá una distribución nueva a partir de una Ubuntu estándar.

Ubuntu-builder

Ubuntu-builder es una herramienta que permite personalizar una distribución Ubuntu. Tiene una interfaz bastante cómoda para su utilización.

No está en repositorio por lo que será preciso añadir un repositorio externo:

add-apt-repository ppa:f-muriana/ubuntu-builder

Jose Enrique Ruiz Navarro

Se actualizará y se instalará.

apt-get update
apt-get install ubuntu-builder

A partir de ahora la podemos encontrar en las aplicaciones de nuestro sistema.

Al abrirla, aparecerá una interfaz que es muy intuitiva.

Ubuntu Builder								
Menú								
3	\$	∔	↑	+	<u> </u>			
Ajustes	Asistente	Importar	Exportar	Construir	Probar			
Desde el disco duro lo V Seleccione un DE/WM V Lista de paquetes								
Obtener Ubu	ntu 🗸	Ed	itar sources.list	s	ynaptic			
			Terminal		scritorio			
		Insta	alar paquetes deb	U	lbiquity			

Para empezar, hay que introducir la .iso descargada desde la web de Ubuntu.

Jose Enrique Ruiz Navarro

			Ubunt	u Builder			×
Menú		_	_				_
~				1		4	<u>•</u>
Ajustes	Asistente	Impo	ortar	Exportar		Construir	Probar
Desde el disco	duro lo 🗸		Seleco	tione un DE/WM	~	Lista d	e paquetes
Obtener Ubunt	tu v		Edi	itar sources.list		S	naptic
			Terminal			Escritorio	
			Insta	alar paquetes de	b	U	biquity

	/bin/bash	×
•	bin/bash 90x25	
<pre># Mounting IS0 # Checking IS0 # Cleaning existing directories # Extracting File System Parallel unsquashfs: Using 8 processors 142709 inodes (145939 blocks) to write</pre>	[DONE] [DONE] [DONE]	
	======================================	

A continuación se eliminarán paquetes para que el tamaño de la distribución se reduzca, para ello la mejor forma de abordarlo es mediante la aplicación synaptic.

Ubuntu Builder							
Menú							
2	\$	₽	1		4	<u>•</u>	
Ajustes	Asistente	Importar	Exportar	C	onstruir	Probar	
Desde el disc	o duro lo 🗸	Sele	eccione un DE/WM	1 ~	Lista d	e paquetes	
Obtener Ubu	ntu 🗸	E	ditar sources.list		Sy	naptic	
			Terminal		Es	critorio	
		Ins	stalar paquetes de	эÞ	U	piquity	

	Synaptic Package Manager	×
File Edit Package Settin	igs Help	
Reload Mark All Upgra	ades Apply Properties Quick filter	
All Installed Installed (auto removat Installed (local or obsolu Installed (manual) Installed (upgradable)	S Package Installed Version Latest Version Description Image: Solution of the second se	
Not installed (residual c	foomatic-db-compressed-p 20120322-0ubunt 20120322-0ubunt OpenPrinting printer support - Compressed PPDs derived POSIX-compliant shell Get Screenshot Get Changelog The Debian Almquist Shell (dash) is a POSIX-compliant shell derived	from th
Sections Status Origin	from ash. Since it executes scripts faster than bash, and has fewer library dependencies (making it more robust against software or hardware failures), it is used as the default system shell on Debian systems.	
Custom Filters Search Results Architecture		×

Después instalaremos spice-client mediante la terminal. Todo lo que se instale o desinstale desde esta consola, lo hará en la distribución creada.

Type 'exit' to quit chroot console
root@stauffenberg:/# apt-get install spice-client

Y para que se ejecute al iniciar la distribución se pondrá en /etc/skel/.profile la siguiente línea:

spicec --host 192.168.1.2 --port 5900

Esto hace que cuando arranque el perfil del usuario ubuntu (por defecto en la live) se inicie la aplicación spice-client.

Una vez que tenemos la distribución preparada, se continúa con el siguiente paso utilizando la opción *construir*. Cuando finalice su construcción, se podrá comprobar que ha creado un fichero con extensión .iso en el directorio /home/ubuntu-builder. También se puede probar la distribución desde el mismo ubuntu-builder, puesto que hay un botón al lado del que indica *construir* para montar la imagen en una máquina virtual usando qemu.

28

Jose Enrique Ruiz Navarro

El proceso se repetirá tantas veces como máquinas virtuales se deseen ofrecer mediante Spice.

La imagen que se ha preparado, es una imagen de instalación, por lo que aparecerá el menú para elegir entre probar ubuntu e instalar ubuntu. Para evitar esto, es necesario preparar la imagen para que se ejecute directamente el modo de prueba.

Instalación de los servicios en el servidor

DHCP

Se instala el paquete:

apt-get install isc-dhcp-server

Se le indica por qué interfaz va a servir, para ello hay que editar /etc/default/isc-dhcp-server:

INTERFACES='eth0'

Se asignará el rango de ip que se vaya a utilizar en la red interna. Pero lo que se diferencia de una configuración normal es el parámetro filename, que será el encargado de mandar el fichero de arranque de pxe:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.40 192.168.1.60;
  option domain-name-servers 8.8.8.8;
  option routers 192.168.1.1;
  option broadcast-address 192.168.1.255;
  filename "pxelinux.0";
  default-lease-time 600;
  max-lease-time 7200; }
```

Jose Enrique Ruiz Navarro

En el caso que el propio servidor no sea el que asigne ip, se añadirá el siguiente parámetro :

```
next-server ipdelservidorpxe;
```

Y finalmente se reinicia el servicio:

/etc/init.d/isc-dhcp-server restart

Tftp-hpa

Se instala el paquete:

apt-get install tftpd-hpa

Se crea el directorio donde estarán todos los ficheros de PXE:

mkdir -p /srv/pxe

A continuación, se configura el demonio en /etc/default/tftpd-hpa:

```
# /etc/default/tftpd-hpa
RUN_DAEMON="yes"
OPTIONS="-1 -s /srv/pxe"
TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/srv/pxe"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS="-secure"
```

Se reinicia el servicio:

Jose Enrique Ruiz Navarro

/etc/init.d/isc-dhcp-server restart

Y finalmente, se comprobará si está en funcionamiento:

root@proyecto:/etc/default# netstat -tupan | grep tftp udp 0 0.0.0.0:<u>6</u>9 0.0.0.0:* 1246/in.<mark>tftp</mark>d

Construcción de imagen para arranque PXE

Se usarán las imágenes que se han construido anteriormente.

Lo primero que hay que instalar es syslinux:

apt-get install syslinux

Después se copia el pxelinux.0 (será el que inicia el arranque a través de la red) a /srv/pxe/, que es el directorio donde tftp localiza los ficheros y los envía:

cp /usr/lib/syslinux/pxelinux.0 /srv/pxe

A continuación, se montará la imagen:

mount -o loop ubuntu-ubuntu.iso /mnt/

Se deberá copiar el kernel que se usa cuando se accede al modo de prueba de ubuntu, estará en el directorio casper:

Jose Enrique Ruiz Navarro

cp /mnt/casper/vmlinuz /srv/pxe/

También se copiará el initrd del mismo sitio:

cp /mnt/casper/initr.lz /srv/pxe/

Como se tendrán varias distribuciones, se creará un directorio para tal cometido:

mkdir -p /srv/os

Dentro del anterior directorio, se creará uno por cada distribución que se quiera ofrecer.

mkdir -p /srv/os/ubuntu

El paso siguiente será copiar el contenido de la iso en el directorio que se ha creado con anterioridad:

cp -r /mnt/* /srv/os/ubuntu/

Y por supuesto el .disk, que montará los ficheros de ubuntu sin que esté instalado en el cliente:

cp -r /mnt/.disk /srv/os/ubuntu/

Se desmonta la imagen:

33

Jose Enrique Ruiz Navarro

umount /mnt

Se creará un directorio de configuración que será leído al iniciar el pxelinux.0 :

mkdir -p /srv/pxe/pxelinux.cfg

Y dentro un fichero llamado default que será leído al arrancar con el siguiente contenido:

```
include menucfg
default vesamenu.c32
prompt 0
timeout 0
```

Este fichero cargará un menú (vesamenu.c32) con el contenido del fichero menu.cgf, ambos se crearán en el directorio /srv/pxe/

Se copiará el que carga el menú:

cp /usr/lib/syslinux/vesamenu.c32 /srv/pxe

Y se creará el fichero menu.cfg con el siguiente contenido:

```
LABEL 1
MENU LABEL máquina Ubuntu
KERNEL vmlinuz
APPEND boot=casper netboot=nfs
nfsroot=192.168.1.2:/srv/os/ubuntu initrd=initrd.lz
TEXT HELP
Boot the Ubuntu
```

Jose Enrique Ruiz Navarro

Proyecto integrado

ENDTEXT

LABEL 2

MENU LABEL máquina Windows KERNEL vmlinuz APPEND boot=casper netboot=nfs nfsroot=192.168.1.2:/srv/os/windows initrd=initrd.lz TEXT HELP Boot the Windows ENDTEXT

Se puede observar que en el menú hay varios niveles, esto es debido a que son dos distribuciones distintas y cuando se seleccione una u otra, se ha de descargar la distribución que esté en dicho directorio. Habrá tantas como máquinas virtuales y distribuciones haya.

El método que se utiliza para transferir la distribución, es mediante NFS, cuya configuración se detallará a continuación.

NFS

Jose Enrique Ruiz Navarro

Se instala el paquete:

apt-get install nfs-kernel-server

En /etc/exports habrá que poner una línea por distribución exportada, por ejemplo si hay dos distribuciones, se pondrán dos líneas:

/srv/os/ubuntu	<pre>*(ro,async,no_root_squash,no_subtree_check)</pre>
/srv/os/windows	<pre>*(ro,async,no_root_squash,no_subtree_check)</pre>

Reinicio de demonio:

/etc/init.d/nfs-kernel-server restart

Problemas encontrados

Spice con Openstack

En la versión mas reciente de openstack (grizzly), se ha empezado a incluir Spice. Podemos ver las líneas de configuración que se deberían incluir

html5proxy_base_url=http://127.0.0.1: 6080/spice_auto.html	(StrOpt) location of spice html5 console proxy, in the form "http://127.0.0.1:6080/spice_auto.html"
enabled=true	Habilitar spice
agent_enabled=true	Habilitar agente spice
keymap=es-es	Distribución del teclado
server_listen=0.0.0.0	Dirección en el que la máquina spice escuchará
server proxyclient address=127.0.0.1	Dirección en el que el proxy se conectará via web

Pero claro, esta implementación es para el navegador, que aparte de usar html5, limita mucho a la hora de conectarse al servidor con un cliente, impidiendo autenticar. Cuando el cliente solicita una máquina a Openstack, recibe una máquina cualquiera, sea o no sea del usuario que la pide, y lo que es peor, exista o no exista el usuario que la pide, por lo que tendría que autenticar con el servidor spice a la hora de realizar la conexión.

Raspberry pi como cliente ligero.

Como es ampliamente conocido, Raspberry pi es un ordenador de bajo costo. Lamentablemente, este dispositivo no dispone de arranque a través de PXE, por lo que no funcionaría en el escenario que hemos construido. Así pues busqué otra

Jose Enrique Ruiz Navarro

Proyecto integrado

alternativa; utilizar PXE por software: gPXE.

GPXE es una implementación de fuente abierta del Preboot Execution Environment (PXE) y cargador de arranque. Puede ser usado para habilitar a los computadores que no tienen soporte para el PXE para que puedan cargar desde la red.

Hay varias formas de iniciar gPXE, sólo es necesario crear el software en :

http://rom-o-matic.net/gpxe/gpxe-1.0.1/contrib/rom-o-matic/.

Una vez que se obtenga la imagen, iniciarla por usb, cdrom o algún otro medio y arrancará como si tuviera PXE:

```
ISOLINUX 3.63 Debian-2008-07-15 Copyright (C) 1994-2008 H. Peter Anvin
Etherboot ISO boot image generated by geniso
boot:
Loading gpxe.krn.....ready.
gPXE initialising devices...
gPXE 1.0.1 -- Open Source Boot Firmware -- http://etherboot.org
Features: AoE HTTP iSCSI DNS TFTP bzImage COMBOOT ELF Multiboot NBI PXE PXEXT
net0: 08:00:27:19:c8:31 on PCI00:03.0 (open)
```

Pero raspberry pi necesita tener un sistema operativo en la tarjeta SD para arrancar y aún no hay gPXE adaptado para tal medio. Indagando, encontré un sistema operativo para raspberry pi con el cliente Spice instalado, que ayudaría a conectarse al servidor, pero con el pequeño inconveniente de conocer previamente la ip del servidor y el puerto de la máquina virtual que se quiera abrir. Esta distribución se encuentra en:

http://rpitc.blogspot.com.es/

Redirección de USB

Como vimos en la comparación de las diferentes tecnologías, una ventaja de spice era la redirección de USB, pero en la versión de qemu actual no está

Jose Enrique Ruiz Navarro

implementada.

8	
	Error iniciando dominio: unsupported configuration: USB redirection is not supported by this version of QEMU
▶ Detall	es
	Cerrar

Conclusión

He abordado tecnologías relativamente nuevas, que están en constante desarrollo, y eso significa que, en algunos casos, son aún inestables o no están completamente implementadas. Así que pienso que es todo un reto aventurarse a trabajar, no sobre una sino sobre dos.

Y es el caso del inicio de mi proyecto, abordé un proyecto que creía que iba a desarrollar sin mucho esfuerzo, o encontraría soluciones sin ningún problema. No sabía que Spice era una tecnología aún tan joven y creía que Openstack era lo suficientemente maduro. Pero descubrí que no es así, ambos están en pleno crecimiento, todos los años actualizan sus versiones con grandes cambios, dejando obsoletas las versiones anteriores. Por ejemplo, grizzly cambia por

completo respecto a essex. Son cambios radicales.

Tengo que sincerarme en esta conclusión; y es que he llegado a perder los estribos, el proyecto de "cliente ligero para Openstack utilizando Spice" me superó, no había documentación alguna. Incluso llegué a preguntar en el foro oficial de Openstack sin éxito. Todo me pareció gris, incluso pensé en suspenderlo. Hasta que me guiaron y me dijeron que no todo es posible. Entonces acoté mi proyecto a un cliente ligero utilizando máquinas virtuales. Me esperancé de nuevo.

Mi conclusión no es sólo educativa, sino moral. Darse por vencido en un problema no soluciona nada; lo empeora. También sé que soy capaz de hacer las cosas que me propongo. Además he aprendido bastantes cosas y espero que el que use este tutorial también aprenda.

Y quedarme con una frase que me animó:

"Cuando la meta es importante los obstáculos se vuelven pequeños"

Jose Enrique Ruiz Navarro

Bibliografía

http://jrs-s.net/2012/10/27/getting-spice-working-in-ubuntu-12-04-1-lts/

http://askubuntu.com/questions/47522/how-to-bypass-try-it-install-screen-when-boo ting-from-usb-live-session-wit

http://www.kraxel.org/slides/2011-linuxtag-spice.pdf

http://spice-space.org/

http://es.wikipedia.com

http://rpitc.blogspot.com.es/

http://web.archiveorange.com/archive/v/Xs3ry9seYcfsUe0G0cCc

http://martinlanner.com/2012/07/28/build-a-pxe-boot-server-environment/