Virtualización I

Miguel Ortuño
Escuela Técnica Superior de Ingeniería de Telecomunicación
Universidad Rey Juan Carlos

Septiembre de 2022

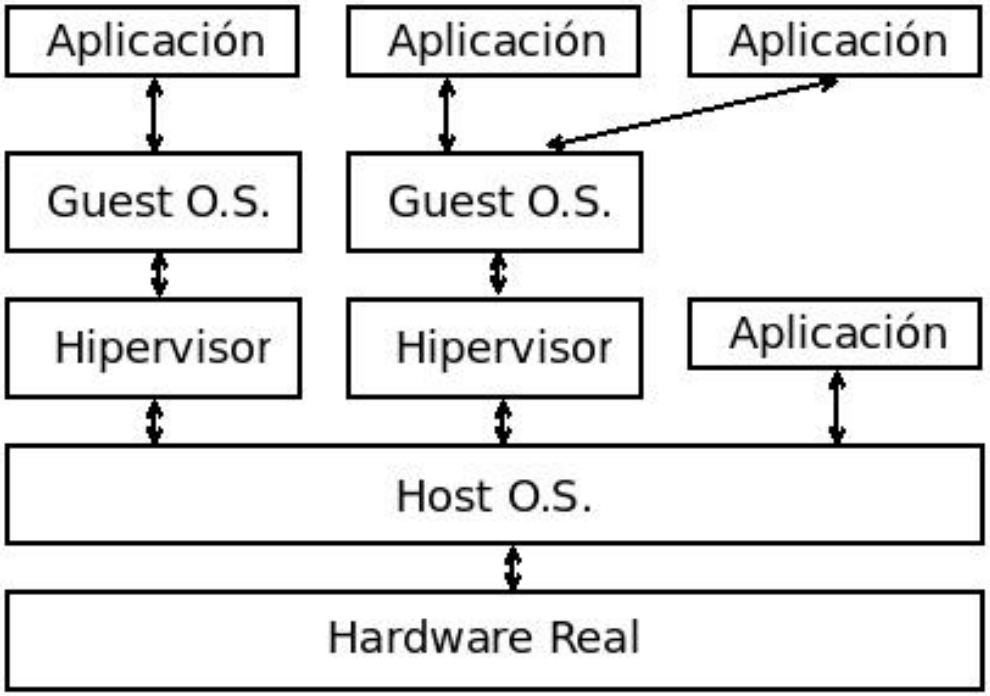
Contenidos

Máquinas Virtuales

Máquina Virtual: Software que crea una capa de abstracción, ofreciendo una máquina diferente a la máquina original
Las máquinas virtuales que nos interesan en administración de sistemas suelen ofrecer a un sistema operativo la percepción de una máquina física

- Las aplicaciones y los usuarios dentro de la máquina virtual se relacionan con la capa de abstracción y no con la plataforma real
- La máquina virtual puede implementar diversos dispositivos virtuales (disco, dispositivos de red, etc) diferentes a los de la plataforma real

- La tecnología sobre Máquinas Virtuales está muy madura. La terminología, no. Es frecuente encontrarse con el distintos nombres para el mismo concepto, o incluso el mismo nombre para cosas distintas
- *Guest*: Sistema Operativo de la máquina virtual
  Host: Sistema Operativo de la máquina real



- La máquina virtual se comporta como una aplicación más en el *host*
- El *guest* percibe la máquina virtual como si fuera hardware real

Uno de los modelos posibles: máquina virtual de sistema

Utilidad de las máquinas virtuales

Tecnología tradicional y actual, con muchas utilidades

- Ejecutar aplicaciones hechas para una plataforma sobre una plataforma diferente: p.e Microsoft Windows sobre Mac OS, Java Virtual Machine
- Ofrecer un entorno seguro donde experimentar (*sandbox*)
  - Docencia
  - Probar aplicaciones en desarrollo
  - Probar aplicaciones o webs no confiables
- Señuelos (*Honeypots*)
- Empresas de *hosting* pueden ofrecer servidores virtuales (alimentación y conectividad redundante, soporte 24/365, etc)

- Respaldo (*backup*) de máquinas enteras, no solo de datos. Ante un pequeño problema o un gran desastre, la máquina virtual se recupera inmediatamente
- Seguridad: Cortafuegos, perímetros de seguridad,...
- Portabilidad: Moviendo un directorio se puede mover la máquina virtual de una máquina real a otra
- Independencia del Hardware, p.e. homogeneizar un conjunto de máquinas diferentes
- ...

Inconvenientes de las máquinas virtuales

Inconveniente principal: pérdida de rendimiento
Aunque no siempre

- La máquina *real* tal vez no existe (p.e. java)
- Existe, pero es una máquina de propósito específico.
  Un guest sobre un host de propósito general puede ser más eficiente

MV de proceso y de sistema

Según su grado de equivalencia sobre una máquina hardware, las máquinas virtuales se pueden clasificar en

- Máquinas virtuales de sistema
  Proporcionan un entorno completo y persistente para ejecutar un sistema operativo y sus procesos
  Ej: VirtualBox, Xen
- Máquinas virtuales de proceso
  Proporcionan una plataforma para ejecutar un único proceso
  - Contenedores
  - Máquina virtual de java, .NET
    Este tipo no lo consideramos dentro del ámbito de la administración de sistemas y no lo tratamos aquí

Emulación Completa o Virtualización Completa

Whole-system virtualization. Se emula memoria, disco y otros dispositivos, también la CPU:
Al emular la CPU, son especialmente lentos. La arquitectura Intel tradicional ofrecía muy pocas facilidades
Permiten que *guest* y *host* trabajen con diferente ISA (*instruction set architecture*)

Ejemplos: QEMU, Bochs.

- Emulan una CPU intel, incluso cuando se ejecutan sobre intel.
- Ambos son libres, disponibles para diversos *hosts*.
- Pueden ejecutar distintos *guest*, pero siempre para intel

Virtualización

- Al virtualizador también se le llama *hipervisor*
- Se emula memoria virtual, disco y dispositivos
  Ejemplo: VMware emula tarjeta de audio SoundBlaster 16 y tarjeta ethernet AMD PCnet II. Cualquier aplicación en el *guest* percibe este hardware
- No se emula la CPU. Por tanto *guest* y *host* tienen que usar la misma arquitectura

VMware. Virtualizador. Software muy maduro. Versiones comerciales y versiones *freeware* (con los años va aumentando el número de versiones freeware)

- VMware Workstation, workstation player. Para host Windows y Linux. Permite crear y ejecutar máquinas virtuales
- VMware Fusion. Similar a VMWare Workstation, para Mac OS
- VMware ESXi. Verdarero Sistema Operativo. Se ejecuta directamente sobre el hardware
- VMware vSphere. Computación en la nube. Basado en ESXi

Parallels Desktop

- Virtualizador para los Mac OS basados en Intel
- *guest* soportados: Microsoft Windows, Linux, FreBSD, Sun Solaris y algunos otros

(los Mac posteriores a 2006, basados en Intel, pueden ejecutar Windows en nativo con Boot Camp)

VirtualBox

- Virtualizador, muy similar a VMware
- Desarrollado por Innotek. Sun compra Innotek en 2008. Oracle compra Sun en 2009
  - Virtual Box Open Source Edition
  - VirtualBox. Software Comercial. Gratuito para uso personal y académico
    Incluye alguna característica adicional, como soporte USB. sATA, iSCSI, Remote Display Protocol (RDP) Server

Paravirtualización

Similar a la virtualización, pero exige un versión ligeramente modificada del *guest*
El rendimiento es normalmente mayor que el de los tipos anteriores
Xen

- Muy extendido
- Hay una versión libre que permite Linux sobre Linux
- Hay versiones comerciales que permiten Windows sobre Windows
- Los drivers están paravirtualizados, son más eficientes. (En un virtualizador, los drivers son drivers hw normales)
- También hay que modificar el *guest* (Xen lo llama *Dom0*)

Virtualización asistida por hardware

También llamada *virtualización nativa*

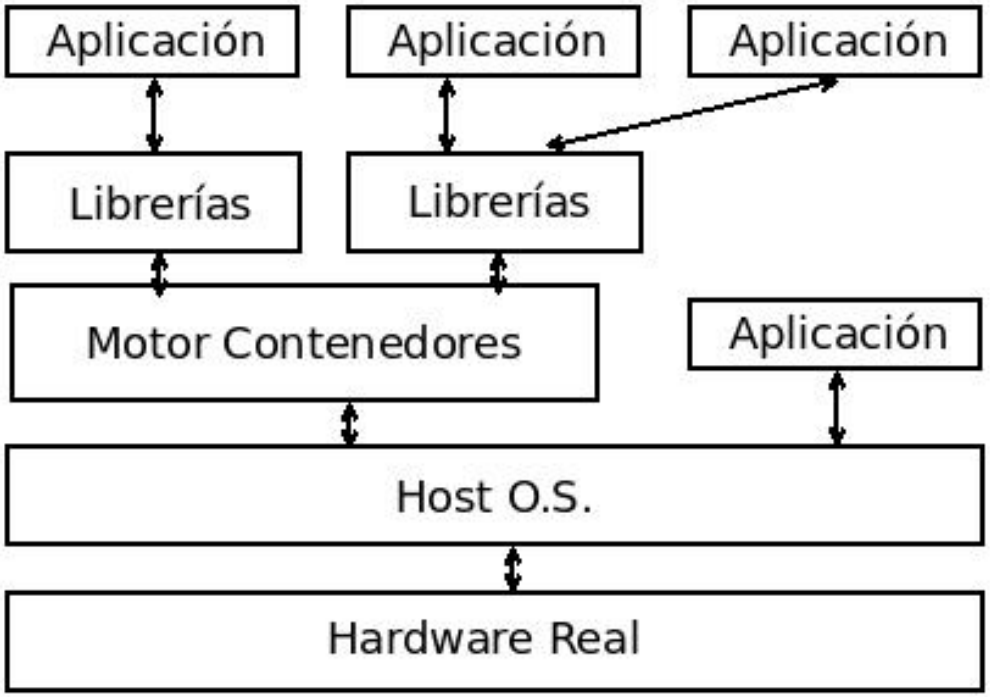
- Es una emulación completa, pero realizada por la CPU con lo que el rendimiento es próximo al nativo
- Exige soporte en la CPU. Para Intel aparece en el año 2006 con KVM: Kernel-based Virtual Machine. Infraestructura para virtualización completa del núcleo de Linux
- Soportado por Xen

Procesadores que lo soportan:

- Intel virtualization (VT-x)
  Pentium 4 6x2, Pentium D 9x0, Xeon 3xxx/5xxx/7xx, Intel Core, Intel Core 2, Intel Quad-Core. Algunos atom (serie Z5xx)
- AMD-V
  AMD con Socket AM2, Socket S1 y Socket F. También procesadores Athlon 64 y Turion 64 a partir de mayo de 2006

Contenedores

- Un contenedor es una encapsulación de una aplicación y todas sus dependencias
- Se pueden considerar una versión aligerada de las máquinas virtuales tradicionales
- Su nombre es una metáfora de los contenedores empleados en el transporte
  - Recipientes de carga que puede transportarse fácilmente en camión, barco o tren sin manipular la mercancía de su interior
  - Revolucionaron la industria en los años 1930
  - Desde los años 1970 son estándares mundiales
- Virtualización a nivel del sistema operativo
- Son máquinas virtuales de proceso, típicamente ejecutan un único proceso, como mucho unos pocos



Contenedores (diagrama simplificado)

Cada aplicacion se ejecuta en su propio contenedor. Cada una de ellas:

- Comparte el mismo sistema operativo
- Tiene la percepción de acceso exclusivo a los recursos
- No percibe a las demás

- No confundir con *web container*, también llamado *servlet container* que es algo completamente distinto:
  - En los servidores web basados en java, un contenedor web es el subsistema que interactúa con los servlets java
- Los contenedores son típicamente un orden de magnitud más eficientes que las máquinas virtuales tradicionales
  - Una máquina virtual suele tardar en arrancar muchos segundos o algunos minutos. Un contenedor, décimas de segundo
  - El rendimiento del proceso en ejecución es casi igual al nativo
- No son incompatibles con las máquinas virtuales tradicionales, al contrario, es muy habitual ejecutarlos dentro de máquinas virtuales
- Los contenedores solucionan el típico problema de *en mi máquina funcionaba*

El desarrollador

- Incluye dentro del contenedor todo lo necesario para que la aplicación se ejecute
- Sabe que la aplicación funcionará de forma idéntica en cualquier entorno (un servidor hardware tradicional, una máquina virtual, un servidor en la nube, un portátil...)

El administrador

- Pueden concentrarse en los recursos de red, sin perder tiempo configurando entornos y dependencias en el sistema

El poco peso de los contendores

- Permite ejecutar docenas de ellos simultáneamente en cualquier máquina
- Facilita su uso en la nube

Los contenedores son una tecnología que está cambiando la forma en la que se desarrolla, distribuye y ejecuta el software.

Historia de los contenedores

- Año 1979. Sistema chroot de Unix V7. En cierta forma pueden considerarse los primeros contenedores, aunque solo encapsulan el sistema de ficheros (no los procesos, ni los usuarios, ni la red...)
- Año 2000. BSD *Jails*
- Año 2001. Linux VServer (año 2001)
- Año 2004. Solaris Containers
- Año 2008. LXC (Linux Containers)
- Año 2013. Solomon Hykes libera Docker como software libre

Docker

Madurez de los contenedores: Docker

- Hasta la aparición de Docker, los contenedores eran una herramienta de nicho. Tenían su utilidad, pero no se puede decir que su uso fuera masivo
- Docker es una herramienta muy fácil de usar y muy eficiente, hace que los contenedores pasen a ser muy populares
- Los organismos que principalmente contribuyen a su desarrollo son, además del *docker team*, Cisco, Google, Huawei, IBM, Microsoft y Red Hat
- Está integrado con las principales plataformas y herramientas: Amazon Web Services, Ansible, CFEngine, Chef, Google Cloud Platform, IBM Bluemix, Jelastic, Jenkins, Kubernetes, Microsoft Azure, OpenStack Nova, Oracle Container Cloud Service, Puppet, Vagrant, VMware vSphere...

Fundamentos de Docker

Docker se basa en la funcionalidad de virtualización ofrecida por el nucleo de Linux:

- cgroups
  El término proviene de *control groups*. Año 2008. Permiten limitar y aislar los recursos consumidos por un grupo de procesos (CPU, memoria, E/S, red...)
- Linux kernel namespaces
  Grupos de procesos que no pueden *ver* los procesos de otros grupos. Quedan aislados los PID, los interfaces de red, las tablas de encaminamiento, el cortafuegos, el nombre de host, los puntos de montaje del sistema de ficheros, la intercomunicación entre procesos y los identificadores de usuario

Union Filesystem

Otra tecnología fundamental integrada en Docker es *Union Filesystem*, también llamado *Union Mounting*

- Docker soporta diversos *drivers* para el UFS: overlay2, aufs, OverlayFS, entre otros
  Emplear uno u otro depende fundamentalmente de la plataforma, el usuario de Docker normalmente no necesita ocuparse de esto
- Las imágenes de los contenedores están formadas por varias capas apiladas
  - Todas las capas son de solo lectura, excepto la última, que es de lectura y escritura
  - UFS permite que cada contenedor perciba todas las capas como un único sistema de ficheros ordinario
  - Pero cada capa de solo lectura puede ser compartida por varios contenedores, de forma que solo la capa de lectura/escritura es exclusiva para cada contenedor

Estas capas diferenciales apiladas representan un enorme ahorro de espacio en el sistema de ficheros
Ejemplo:

- 10 imágenes similares de una máquina virtual de 2 Gb ocuparán necesariamente 20 Gb
- 10 imágenes similares de un contenedor pueden ocupar 2.2 Gb

Inconvenientes de los contenedores (1)

Todos los contenedores comparten el mismo kernel con el host
Esta es la principal ventaja (son ligeros) pero también el principal inconveniente (no son muy seguros)

- Una vulnerabilidad o un kernel panic en un contenedor afecta a toda la máquina
- Cargar un módulo del kernel en el contenedor es cargarlo en el host
- Existe el riesgo de ataques DOS (Deny of Service)
- No hay un espacio de nombres propio para los usuarios en el contenedor. Si un usuario es root en el contenedor y consigue salir del contenedor, es root en el host

Inconvenientes de los contenedores (2)

Para poder lanzar un contener son necesarios privilegios de superusuario en el host. Es necesario

- O bien ser root o usar sudo
- O bien pertenecer al grupo *docker*, lo que resulta equivalente

Previsiblemente este problema se resolverá en futuras versiones de Docker, con un espacio de nombres propio para los UID

Inconvenientes de los contenedores (3)

Otros elementos compartidos por host y contenedores:

- Los dispositivos: discos, tarjetas gráficas, de sonido...
- La hora
- El anillo de claves del núcleo

Otros tipos de virtualización

Como acabamos de ver, hipervisores, paravirtualizadores, virtualización nativa y, recientemente, contenedores, son las herramientas de virtualización más habituales en administración de sistemas

Pero hay muchas técnicas posibles, entre otras

- Máquinas virtuales cooperativas
- User Mode Linux

Máquinas Virtuales Cooperativas

- *Cooperative Virtual Machines*. UML y similares
- Término no demasiado extendido, acuñado para coLinux
- Dos sistemas operativos en paralelo acceden al Hw
- El Hw no se virtualiza
- No muy usado

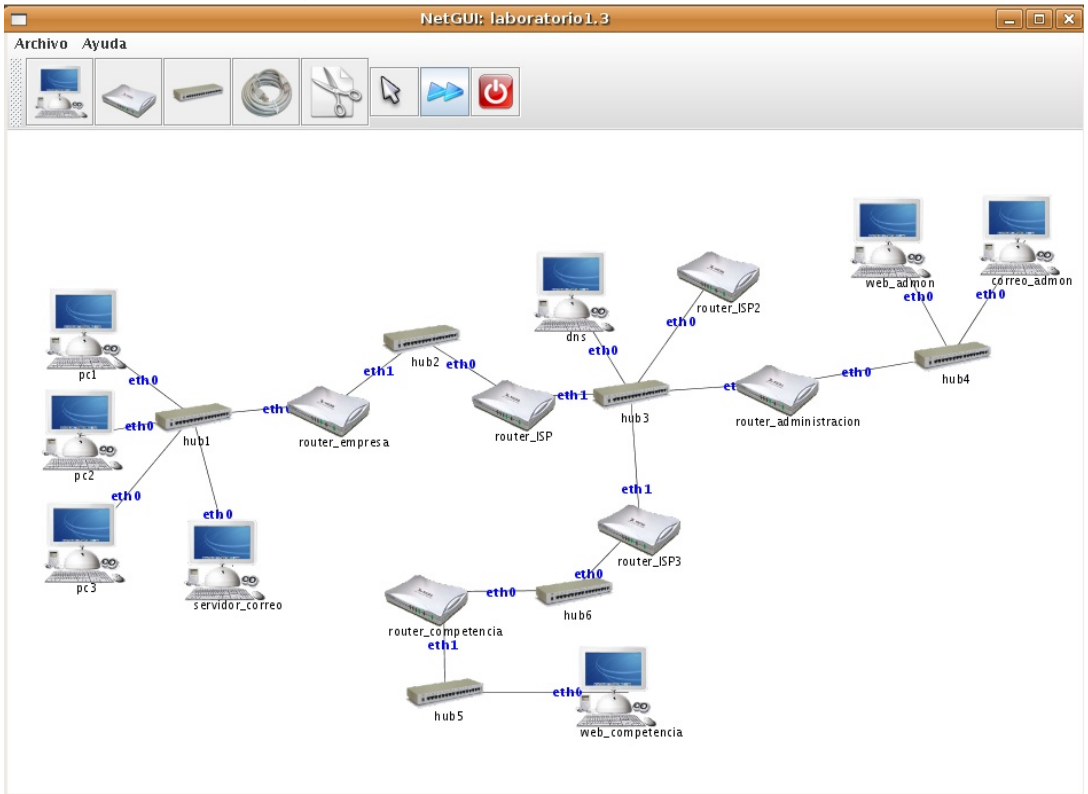
User Mode Linux

- UML. No confundir con *Unified Modeling Language*
- Es un tipo de máquina virtual muy diferente a las anteriores: Un nucleo Linux ligeramente modificado para ejecutarse como un proceso de usuario sobre otro nucleo Linux
- Permite ejecutar diferentes versiones de Linux sobre diferentes versiones de Linux
- Diseñado para Intel, hay versiones para IA-64 y PowerPC
- Los dispositivos del *guest* no están virtualizados. Por tanto en el *guest* se percibe el hardware real
- No muy usado

Netkit

- Entorno basado en UML para emular redes: PCs, routers, conmutadores
- Software libre, desarrollado por la Universidad de Roma

NetGUI



- *Front-end* gráfico para Netkit
- Desarrollado en GSyC

coLinux, AndLinux

coLinux:

- Año 2004. Basado en UML
- Actualmente en desuso
- Versión del núcleo de Linux que se ejecuta sobre otro S.O, como Windows

AndLinux

- Distribución basada en Ubuntu con versión del núcleo de Linux para ejecutarse sobre Windows 2000, XP, 2003, Vista, 7 (solo las versiones de 32 bits)
- Usa coLinux
- Algunos servicios van sobre Windows nativo: Servidor de X Window (Xming), servidor de sonido (Pulse Audio)

Windows Subsystem for Linux (WSL)

WSL 1

- Año 2016
- Capa de compatibilidad que permite hacer funcionar ejecutables Linux sobre Microsoft Windows. Análogo a Wine, pero al revés
- Mucho más ligero que una máquina virtual. El hardware no está virtualizado, el subsistema Linux comparte recursos con los procesos Windows
- El núcleo de Windows se modifica para ejecutar directamente procesos Linux

WSL 2

- Año 2019
- Incluye un verdadero núcleo de Linux
- El hardware se virtualiza, pero con una máquina muy ligera (ligera como p.e. un contenedor)
- Compatible con practicamente cualquier ejecutable Windows, no solo x64, también x32
- Permite usar módulos Linux convencionales

En ambos casos (WSL 1, WSL 2)

- Es una herramienta estándar en Windows, desarrollada por Microsoft. Gratuita (aunque no libre)
- Hay varias distribuciones Linux preparadas para ejecutarse en este entorno, siende Ubuntu la más habitual
- Permite usar aplicaciones gráficas mediante un servidor X Window para Microsoft Windows.
- El objetivo es que los programadores puedan desarrollar código para Linux en sus máquinas Windows con mejor rendimiento y mayor comodidad que en una máquina virtual tradicional. No poner servicios en producción
- Podríamos usarlo para esta asignatura, parece funcionar bastante bien. El problema es que es un Linux un poco *raro*, con algunas peculiaridades
  - A diferencia de un Linux dentro de una máquina virtual tradicional, prácticamente indistinguible de un Linux en nativo

Técnicas sin virtualización

Instalación, reconfiguración y replicación automática, independecia de la plataforma, portabilidad... son características deseables en una buena administración de sistemas

- Todas ellas pueden conseguirse mediante virtualización
- Pero la virtualización no es la única forma. Hay multitud de técnicas de administración alternativas que también ofrecen estas cualidades
- Un buen administrador de sistemas evaluará lo más adecuado para cada caso

Veremos a continuación alguna de estas técnicas

Jaulas chroot

- Se cambia el directorio raiz que percibe un proceso, (y sus hijos) de forma que no puede acceder fuera de cierto directorio.
- No se aisla el acceso a otros procesos, memoria, CPU, red u otros dispositivos

Simuladores

- Simulan algunas características del comportamiento externo de un sistema. P.e. simuladores de red (*GloMoSim*, *JSIM*, *ns-2*, *OPNET*, *OMNet*, etc)
- Los mal llamados *simulador de Zx-Spectrum para PC*, *simulador de Commodore 64 para PC*, etc, no son simuladores. Son emuladores completos.

Capas de Compatibilidad

- Wine. Reimplementación de la API de Win16 y Win32 para sistemas operativos basados en Unix bajo plataformas Intel. Permite ejecutar algunas aplicaciones para Windows en Linux. Cedega es un *fork* comercial de Wine
- Cygwin. Año 1995. Entorno para portar software POSIX a Windows, compuesto por:
  1. DLL que ofrece la funcionalidad de las llamadas al sistema de Linux
  2. Colección de herramientas habituales en sistemas Unix

  Siempre es necesario recompilar las aplicaciones

Implementación de protocolos de red

- En redes Windows los directorios e impresoras se exportan mediante los protocolos smb/cifs NetBIOS.
  Samba es una implementación de estos protocolos, permite usar máquinas Unix en redes Windows
- En Unix los directorios se exportan normalmente mediante NFS. Hay implementaciones de NFS para Windows. Permiten acceder a directorios Unix desde máquinas Windows
  - En Unix las impresoras se exportan normalmente mediante LPD (*Line Printer Daemon Protocol*). Estándar basado en TCP, RFC 1179.
    Windows entiende este procolo, no hace falta software adicional

Clonación

Permite replicar el disco de una máquina, y con ello todo su S.O. , configuración, aplicaciones y datos

- Normalmente exige máquinas idénticas
- Las herramientas suelen poder clonar cualquier máquina, con independencia de su S.O.
  - Clonezilla. Libre, multiplataforma
  - Norton Ghost. Soft propietario para Windows
  - Acronis True Image. Soft propietario para Windows
  - Partition Saving. Freeware para Windows
  - Partimage. Soft libre, basado en linux, permite clonar cualquier S.O. Viene incluido en *SystemRescueCd*, una distro *live* orientada a recuperar y reparar un sistema
  - SystemImager. Soft libre para Linux.
    Uso típico: Se instala un PC, el *cliente de oro*. La imagen se almacena en el servidor. Esta imagen de distribuye por la red (local), clonando el PC. Si es necesario recuperar una imagen, solo se distribuyen los cambios

Instalación automática del S.O.

Sistema que contesta automáticamente a las preguntas que hace un SO en su instalación.

- *preseed* (debian)
- *kickstart (Red Hat)*
- *nLite* (Windows XP)
- *vLite* (Windows Vista)

Instalación automática de aplicaciones web

Librerías de scripts que instalan aplicaciones web

- Muy usadas por los servicios de hosting
- El interfaz de usuario suele ser via web
- Las aplicaciones que soportan suelen ser aplicaciones para el web
- Ejemplos: Softaculous, Installatron, Fantastico

Herramientas de administración centralizada

Herramientas que se encargan de que los ficheros de configuración se *mantengan* en cierto estado (sin necesidad de preparar scripts que busquen las inconsistencias y las corrijan)

- cfengine
  Herramienta tradicional, muy potente. Manejo de cierta complejidad
- landscape
  Para ubuntu. De pago
- spacewalk
  Para Red Hat y CentOS
- Puppet
  Herramienta muy popular. Basada en Ruby. Algo pesada
- Ansible
  Muy ligera, no necesita demonio en el cliente, solo ssh. En auge

- MSCCM (Microsoft System Center Configuration Manager) Herramienta nativa en Windows para administración centralizada
- Chef, Bcfg2, otras alternativas