CentOS: Virtualization

Tim Verhoeven (tim.verhoeven.be@gmail.com)

Who am I ?

- System administrator at DNS.be
- CentOS contributor :
  - Support (IRC, mailing lists)
  - QA Tester
  - Wiki maintenance
  - Presenter

Agenda

- Virtualization ?
- Types of OS virtualization
- Virtualization on CentOS
- Real life Xen use
- Cobbler & Koan
- Func

CentOS ?

- The short version :
  - Community version of a PNAELV (Prominent North American Enterprise Linux Vendor) Enterprise distribution
  - The aim is 100% binary compatibility
  - Enterprise means :
    - Long lifecycles (7 years)
    - Longer timeframe between releases 18-24 months
    - Stable ABI/API

Virtualization ?

- It is a very broad term and a buzzword
- Can happen at different layers
  - Network virtualization
  - Storage virtualization
  - System virtualization
  - ...
- Common goal is to increase manageability and flexibility

Why virtualize ?

- Isolation (a.ka. security)
- Consolidation (a.k.a. save money)
- Continued use of a legacy application
- Development and testing

Common virtualization techniques

- Emulation (bochs, qemu, ...)
- Full virtualization (VMware, VirtualBox, qemu)
- Para-virtualization (Xen)
- Hardware-assisted virtualization (Xen, KVM, VMware, VirtualBox)
- OS-Level virtualization (OpenVZ, Linux-VServer)

Emulation

- Create hardware in software
- Pro's
  - Support for non-native platforms (PPC on i386)
  - Runs any OS that supports the emulated hardware
  - Useful for low-level debugging
- Con's
  - Very, very slow

Full virtualization

- Let the virtualized system use the host CPU directly
- Problem: privileged instructions are not allowed in user mode
- The hypervisor (layer between hardware and virtual system) needs to handle the priviliged instructions
- Scan for problematic instructions and add a trap to the hypervisor

Full virtualization

- Pro's
  - Decent speed
  - Run any OS that the emulated hardware supports
- Con's
  - x86 instruction set is hard to virtualize
  - Hardware still needs to be emulated

Para-virtualization

- Modify the guest operating system kernel to work with the hypervisor
- The guest system informs the hypervisor when privileged calls need to be made
- The hypervisor provides virtualized devices for the guest
- The guest has special drives for these virtual devices

Para-virtualization

- Pro's
  - Very fast
  - Allows for interaction between host and guest
- Con's
  - Requires modification to the guest OS kernel

Hardware assisted virtualization

- Modern Intel and AMD CPU have extra instructions to help in virtualization
- VT-X for Intel, AMD-V for AMD
- Allows the hypervisor to handle privileged instructions more easily

Hardware assisted virtualization

- Pro's
  - Fast
  - Simpler hypervisor (e.g. KVM)
- Con's
  - Hardware still needs to be emulated
  - Requires modern hardware

Virtualization in CentOS

- CentOS 5 currently supports :
  - Xen (CentOS base)
  - KVM (CentOS extras)

Xen in CentOS

- Fully integrated in CentOS 5
- CentOS 5.1 includes Xen 3.1 hypervisor
- Supports CentOS 4.5+ and CentOS 5.0+ as paravirtualized guests
- If your CPU supports VT-X or AMD-V then you can run unmodified operating systems (older CentOS versions, Windows)

Howto install Xen in CentOS

- Select “Virtualization” during installation
- Or use “Add/Remove Software” to add it later
- The system will reboot by default in Xen-mode

Management tools

- GUI
  - virt-manager
  - vm-applet
- Console
  - xm
  - virsh
  - virt-install

Demo time !

KVM in CentOS

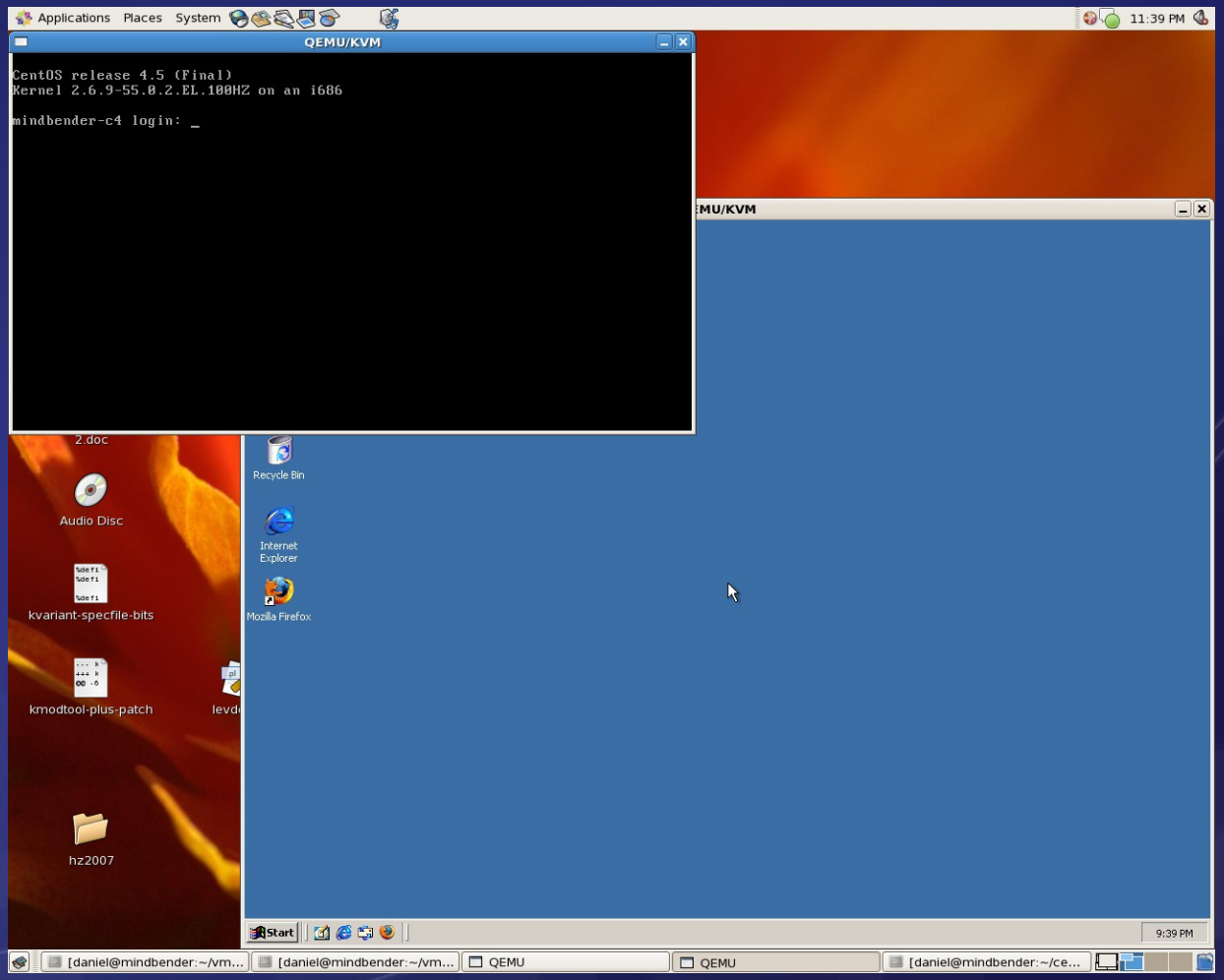
- “yum install kvm kmod-kvm”

KVM – The basics

- KVM requires VT-X or AMD-V
- Hypervisor is a kernel module
- Uses qemu for device emulation

KVM Quickstart

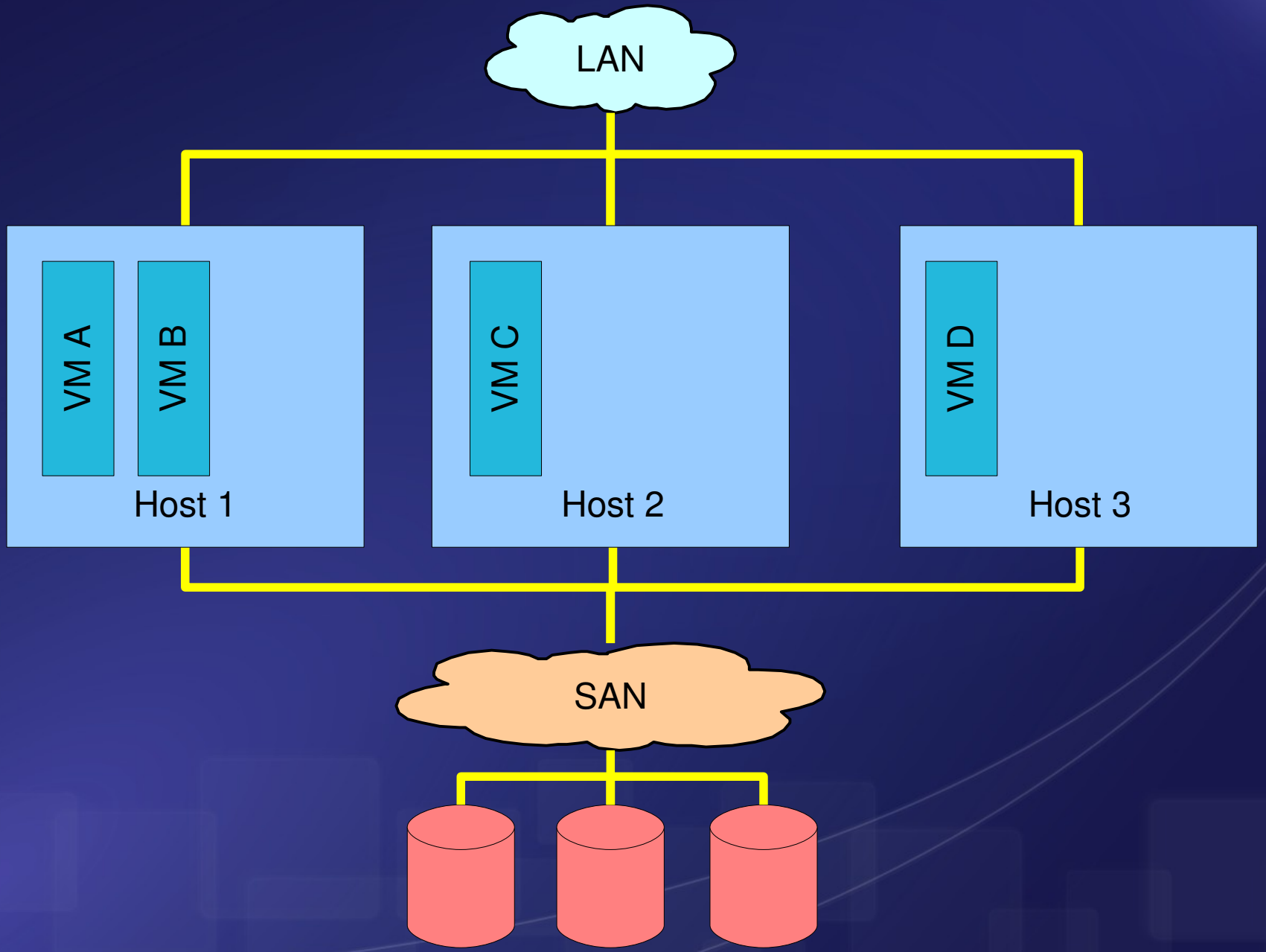
- “qemu-img create -f qcow2 centos5-inst.img 4G”
- “qemu-kvm -hda centos5-inst.img -cdrom boot.iso -boot d”



Real life Xen uses

- Single machine :
  - Testing, development, demo, ...
  - The standard tools work
- Multiple machines :
  - Consolidation, HA, ...
  - More tools are needed :
    - Cobbler & Koan
    - Func

Basic architecture



Variations on the same theme

- Fibre Channel, iSCSI, AoE
- NFS, GFS, ...
- Bridging, NAT'ing, ...
- Files, LVM, Clustered LVM, EVMS, ...
- ...

Step 1 : Installation - Cobbler

- Rapid installation server
- http://cobbler.et.redhat.com/
- Consists of 2 parts :
  - Cobblerd : daemon on install server + WebUI
  - Koan : tool to do reinstalls and virtualized installs
- Expandable :
  - Kickstart templating using Cheetah
  - API using Python or XML-RPC

Installation

- Available in EPEL or rebuild the SRPM
- Edit /var/lib/cobbler/settings
- Run “cobbler check” and fix all errors
- For the WebUI :
  - https://fedorahosted.org/cobbler/wiki/CobblerWebInterface
- Import a distribution “cobbler import ...”

Demo time !

Installing a VM using Cobbler

- Use Koan :
  - Install the RPM on all hosts
  - koan --virt \
    --server=127.0.0.1 \
    --profile=CentOS-5.1-xen-i386 \
    --virt-name=CentOSTest

Demo time !

Kickstart templating

- Create templates in /etc/cobbler
- 4 levels of flexibility :
  - $var to include variables from Cobbler
    - Standard variables from Cobbler
    - Self defined metadata (--ksmeta)
  - Use SNIPPET::file to include simple files
  - Use Cheetah for flexible templating
  - Use Python code for the crazy stuff

Step 2 : Managing - Func

- https://fedorahosted.org/func/
- Like Distributed shell, clustered SSH, ...
- But
- Using Python and XML-RPC over SSL
- Module based (commands, rpm, yum, libvirt, ...)
- Flexible output (standard, JSON, XML)
- Use API to use Func inside other applications

Installation

- Available in EPEL or rebuild the SRPM
- Start certmaster on master server
- Edit /etc/func/minion.conf on slaves
- Start funcd on slaves
- Sign certificates from the slaves on the master using certmaster-ca
- Use func to do stuff ...

Demo time !

To wrap up ...

- CentOS Virtualization SIG :
  - Mailing list : centos-virt on http://lists.centos.org/
  - Wiki : http://wiki.centos.org/SpecialInterestGroup/Virtualization
- The pieces are coming together !

Questions ?