

Whats up at the virtualization/ emulation front?

Christian Horn / globalc

Easterhegg 2010

This file is licensed under the Creative Commons Attribution-ShareAlike License
To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0>

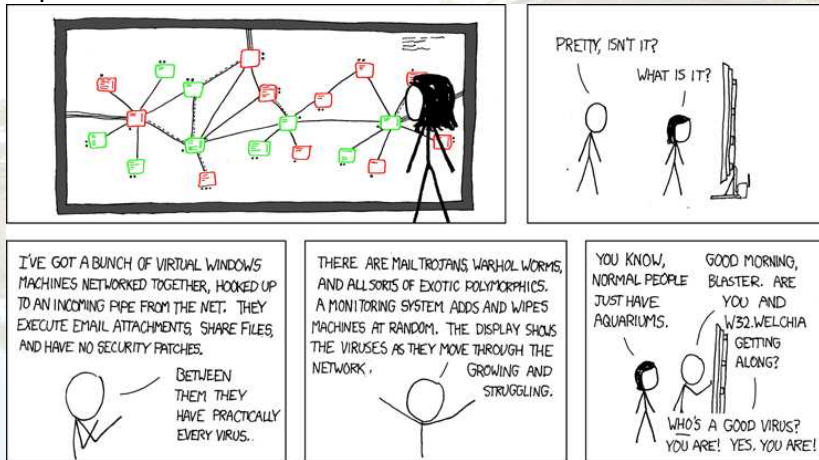
on the content

notice

- This is virtualization in a nutshell as i see it.
- I will skim through the technology in chronological order, basic mechanisms will be explained along that way.
- Will just name the most important software from my point of view, have to simplify things to some level.
- If you have questions, don't wait to ask them! If the question will take too long to answer or take us too far off topic, we'll move on and you can talk to me afterward.

Why is it interesting? Potential hackvalue.

<http://xkcd.com/350/>



start
oo

big iron
●ooooo

void
o

bochs
o

vmware
ooooo

lpar
o

qemu
oo

xen vb
oooo oo

kvm
oo

summary
ooo

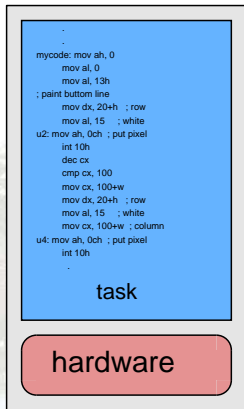
future
oooo

end
o

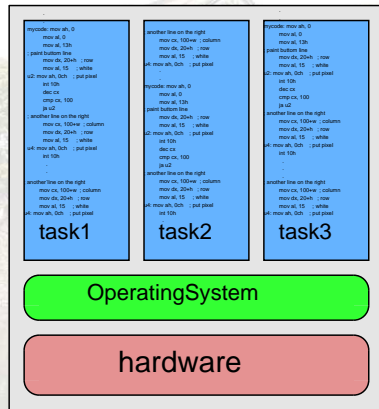
IBM System/360 Model 67

- 24bit mainframe shipped since 1966
- CP/CMS works as native hypervisor (so sits directly on metal) comparable to xen-kernel or vmware ESX
- features like memory protection appearing, time-sharing comes up (in past single big programs), operating systems managing multiple tasks
- funny note: first versions of CP/CMS without copyright-notes so free for publice use :)

jump single tasks -> OSs with multiple tasks



Code running directly on classic computers (think calculators)

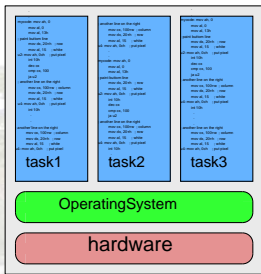


Code running as tasks in an OS which runs on the hardware

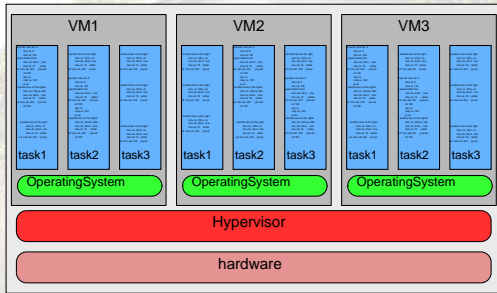
Also starting.. virtualization!

- Why virtualization? To keep old software running, the new hardware allows this now
- accomplishing full virtualization of hardware: all features used by software like i/o, cpu functions, memory etc. are provided in virtual machines -> multiple OSs can now run virtualized

development of hypervisor virtualization



Single OS running multiple tasks



Hypervisor running multiple OS with multiple tasks

start
oo

big iron
oooo●o

void
o

bochs
o

vmware
ooooo

lpar
o

qemu
oo

xen vb
oooo oo

kvm
oo

summary
ooo

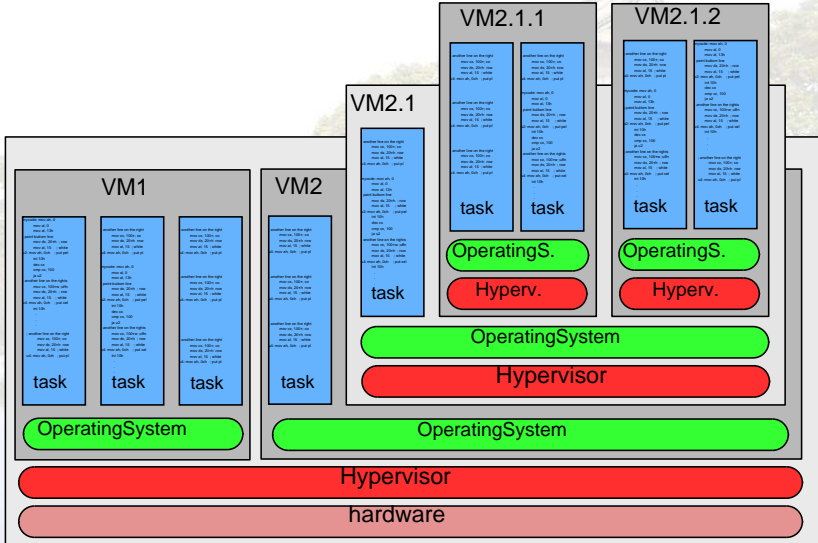
future
oooo

end
o

IBM System/360 Model 67 cont'd

- paravirtualization supported, aka “DIAGNOSE code” in IBM-terms
- System/360 later evolved into System/370, System/390, zSeries, System z9 and current System z10
- current OSs are z/VM or linux. A z/VM acting as hypervisor can run other z/VMs virtualized, even cascading possible
- nowadays hercules emulates z-architecture, even architectures that dont exist in reality

cascading virtual machines



the void

- ...and then during the 1980s and 1990s virtualization is abandoned, the trend is to spread apps over many small lowcost computers
- x86 architecture rises as industry standard, but lacks functions to support virtualization, grows fast and stays backward compatible

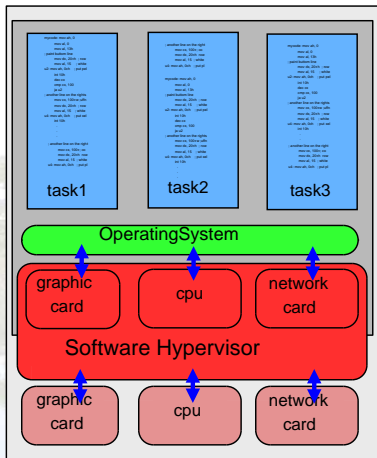
Bochs (spoken: box)

- created 1998, bought by MandrakeSoft and released under LGPL in 2000
- no virtualization, pure emulation of x86-hardware => relatively slow
- often used for OS-development when accurate emulation is required and speed doesnt count
- parts reused in other projects like qemu and kvm
- runs on: linux, macosx, bsd, win
- as guests: most x86-systems like linux, bsd, dos, win

VMware on Desktop

- 1999 VMware publishes its 'Workstation 1.0': focusing on desktops
- among fastest emulators, closed source, HVM required
- features: usb, smp (x16), complex snapshots management, seamless mode integrates guests apps in hosts windowmanager (aka unity), Directx8/9 for winguests
- currently around: Player (free, now also creates VMs, no complex snapshots), Workstation (feature-rich Player, payware), Fusion (Workstation for Mac)

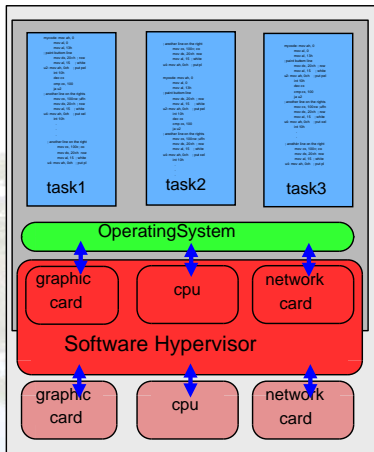
software emulators ways of access



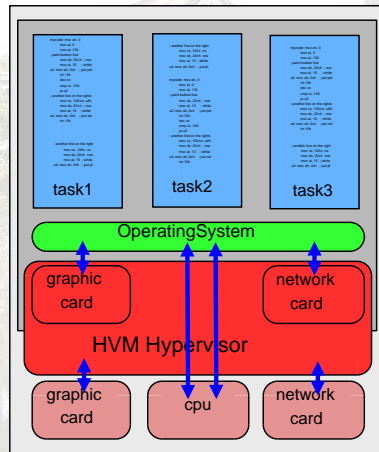
pure software emulator

i.e. Bochs

software emulator vs. virtualizer access



pure software emulator
i.e. Bochs



CPU virtualized
i.e. XEN / AMD-V

VMware Server software

- in 2001 introducing VMware Server (lives on host-OS just as another process) and ESX-server (running pure ESX-kernel on hardware)
- ESXi: by VMware now recommended over ESX
- nowadays well established in industry

Why is VMware so widely used?

- Was first one on market, software long time around
- unique features: low skills required
- most trusted by companies, together with m\$ and citrix
- Contrary to i.e. xen/para, OpenVZ etc. the VMs can be handled like usual servers (pxe-deployment etc.) - customers like that

features...

- overcommit ram/disk, DeDUP for backups/VM snapshots,
- famous for conversion-tools, HA-functions, management GUIs
- virtualizing complex network-setups, bandwidth limitation
- spread VMs as load/energy requires (DRS/DPM)
- live-migrate among different storages (Storage VMotion)
- run shadow-VM to take over failing VM (Fault Tolerance)

IBM power lpar

- 2001 IBM brings VMs also to Power arch as LogicalPARTitions
- hypervisor mainly in hardware, special LPARs for I/O “VIO”
- virtual scsi-traffic among LPARs possible
- live migration == Live Partition Mobility (LPM)
- Power6 VMs can be live migrated to Power7
- ActiveMemorySharing AMS allows in/decreasing RAM of LPARs and overcommit to disc, Active Memory Expansion (AME) compresses RAM in VMs, but nothing like KVM does with KSM

The Qemu generation

- on Mar/23/2003 Fabrice Bellard (ffmpeg, tcc) announces an “x86 emulator” for x86 and PowerPC Linux hosts: “Its main goal is to be able to run the Wine project on non-x86 architectures.”
 - mode linux-user runs linux-binaries across different archs
 - mode system-emu provides a full system
- emulated architectures: x86, ARM, SPARC, MIPS, m68k (and some others like alpha in less usable states)
- snapshots of virtual machines possible

Qemu features

- much hardware emulated: haddisks, cdrom, different nics, sound chips, usb-devices, smp cpus, graphic cards
- also using hosts usb-devices possible
- qcow2 growing diskfiles (think of sparsefiles), http-blockdevices
- speedups possible by using kqemu, doesnt require HVM-cpu-technology!
- most important opensource-project in virtualization area, many reuses in xen, kvm, virtualbox, maemo
- >50% of coders also contribute to linux-kernel

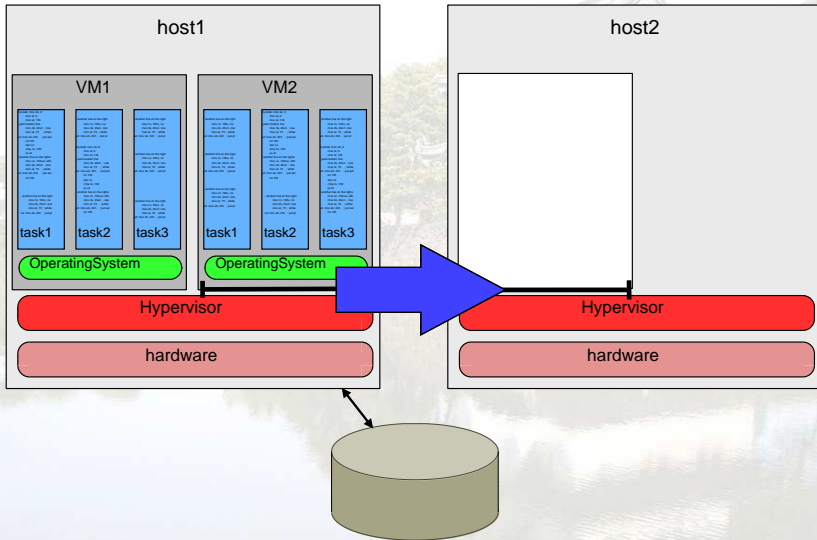
Xen

- in 2003 released by Univ. of Cambridge, nowadays owned by Citrix
- hypervisor, separated in vm 'dom0' for control and real VMs
- 2 modes: paravirt. (OS modified for operation) and fullvirt. (runs unmodified OSs)
- using QEMU-code for emulation of devices
- similar: commercial Citrix (adds GUI/management), sun xVM (think of xen with ZFS and Gui), Hyper-V (since 2k8r2 live-migration, p2v transfer)

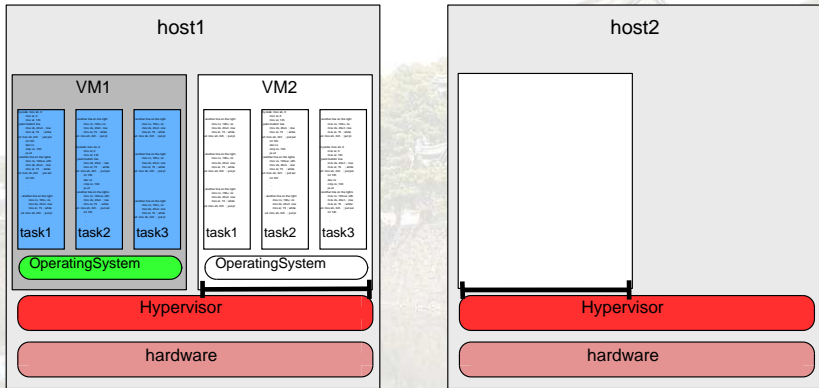
Xen features

- features: fastest soft for paravirtualization, live migration, nowadays also puts newer cpus into sleepstates to save power, GPL paradrivers, pcidevice delegation, vista++ have para-extensions for hyper-v, vmgl
- problems: fewer features than linuxkernel, code not in vanilla kernel, code heavy for adoption

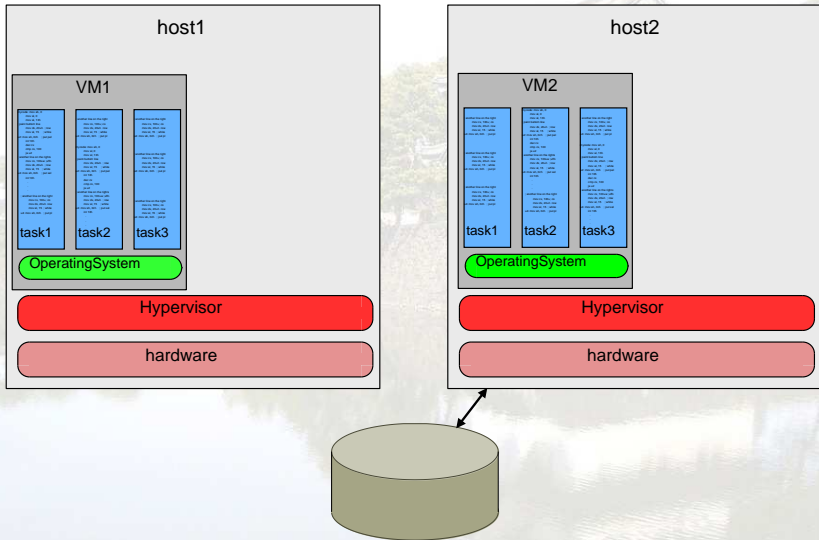
live-migration: memorytransfer of running vm



live-migration: freezing IO and copy of remaining memory



live-migration: unfreezing IO, VM is moved



VirtualBox overview

- in 2007 Innotek (now part of SUN) releases VirtualBox OSE (OpenSource Edition) under GPLv2
- SUN sells support for the partly closed 'Personal use and Evaluation' (PUEL)-version, also adds features
- supports as hosts OS/2 and Solaris (besides usual Linux, MacOS X, Windows)
- fastest free emulator for computers without HVM-function in the CPU, comparable to vmware
- code partly from qemu, nowadays much faster development than qemu

VirtualBox features

- features: rdp-server, usb 1.1/2.0, snapshots, HVM-usage if provided, seamless mode integrates guests apps in hosts windowmanager, clipboard, opengl-drivers/Directx8/9 for winguest on linux-host, smp, livemigration to different host-OS/cputype (teleportation)
- emulates disks backed by files, iscsi, vmdk-files (vmware)
- on roadmap: paravirt windows drivers

KVM kernelbased virtual machine

- *2006 in kernel from Qumranet, later bought by RedHat
- OpenSource
- just a linux kernelmodule utilizing cpus HVM-functions => all features like powermanagement already in place, each virtual-cpu is a thread in the host
- runs unmodified guests, qemu-code emulates devices in userland
- paradrivers for windows & linux for better performance
- AlacrityVM fork for better throughputs& lower latencies

KVM features

- cpu/memory overcommit, live migration, hotplug of cpu/block/nic/pci
- SR-IOV (single root i/o virt.): native pci-e device sharing
- KSM (Kernel Shared Memory): merges equal memory-pages into “shared pages”, think of filesystem hardlinks or Dedup => 600 VMs on a host with 48 cores and 256GB RAM
- pci dev. assignment: requires VT-d or IOMMU hardware (no memory overcommit, no migration, no graphic cards)
- Android & MacOSX virt. possible :)

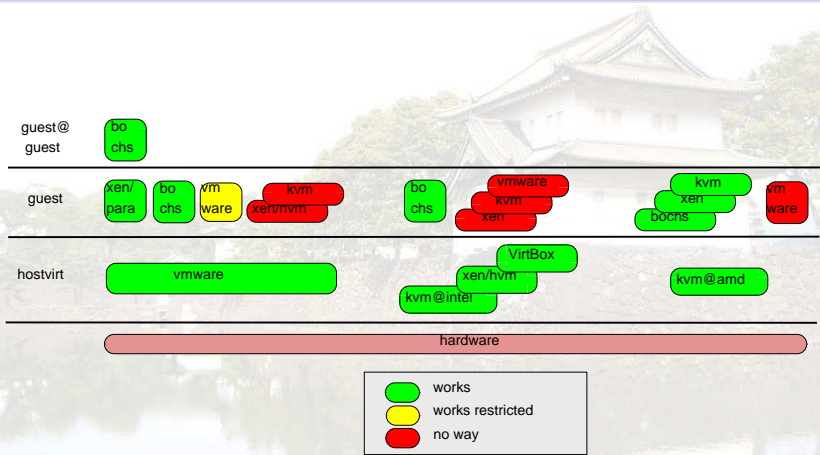
important others that didnt make it to own slides

- for one OS: wine, dosbox
- on one hostkernel: OpenVZ, bsdjails (hierarchical, networked), chroot
- guest having own dependent kernel: Solaris Zones, User Mode Linux, Lguest
- FAUmachine (Hardware Emu/Virt, also emulates faulty hardware)
- whole hardware emus: UnixAmigaEmulator, hercules (Z-arch), SIMH (pdp, vax)
- CooperativeLinux: sharing hardware with windows-kernel at same time

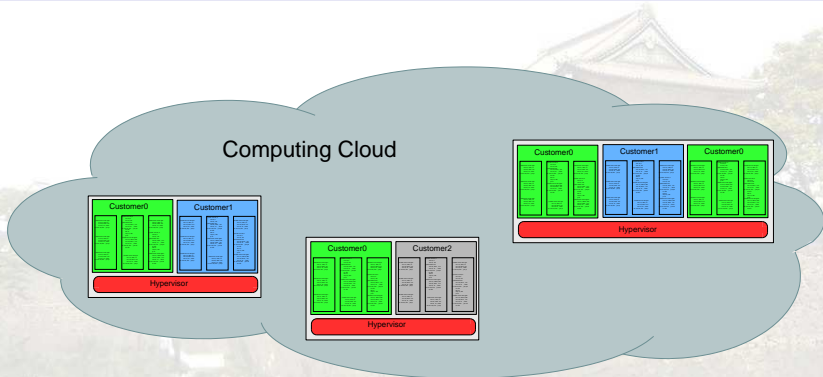
The reasons for virtualization

- security: PS3, separation of daemons into VMs
- consolidation: collecting many VMs on one metal
- provide rare hardware: amiga via uae, winnt@vmware
- rapid provisioning: vm-copy
- fault tolerance: vm boots on several hosts, vm-migration to different hardware

What about cascading today?



cloud overview



Customer0s Virtual machines



Customer1s Virtual machines



Customer2s Virtual machines

cloud topics

- customers apps and their interaction have to be cloud-compatible as cloud is designed for horizontally scaling apps
- cloud-provider trustworthy enough for customers data?
- protection of VMs against each other becomes more important
- why not using clouds: security, we are cheaper, clouds not yet ready

clouds: whats coming up

- libvirt as wrapper over xen, kvm, qemu, openvz, vmware, ibm power
- Open Virtualization Format (OVF), open specs for storing virtual machines
- emerging open frameworks for management:
 - Eucalyptus
 - RedHats RHEV (version 3.x to work without windows)

start
oo

big iron
oooooo

void
o

bochs
o

vmware
ooooo

lpar
o

qemu
oo

xen
oooo oo

vb
oo

kvm
oo

summary
ooo●

end
o

also in a not too far future...

- virtualization on embedded/mobile for security
- linux process freezing/unfreezing
- vms get used like appliances (use while still downloading, boot from internet)
- live-migrations advancing, proxmox already does
kvm-migrations over ssh
- getting a common layer spanning all x86-virtualizers? So vms are movable/migratable i.e. xen<->kvm



Hoping you found the next toy to play with?

<http://fluxcoil.net>

chorn@fluxcoil.net