Virtualization with KVM

FOSDEM 2012

Paolo Bonzini
Sr Software Engineer, Red Hat
February 4, 2012

Outline

- KVM: what made it successful?
- What came out of it?
- How do you use it?
- What will the future bring?

What is KVM?

From: **Avi Kivity <avi@qumranet.com>**
To: linux-kernel <linux-kernel@vger.kernel.org>
Subject: [PATCH 0/7] KVM: Kernel-based Virtual Machine (v2)
Date: Mon, 23 Oct 2006 15:28:48 +0200

The following patchset adds a driver for Intel's hardware virtualization extensions to the x86 architecture. The driver adds a character device (/dev/kvm) that exposes the virtualization capabilities to userspace. Using this driver, a process can run a virtual machine (a "guest") in a fully virtualized PC containing its own virtual hard disks, network adapters, and display.

Using this driver, one can start multiple virtual machines on a host. Each virtual machine is a process on the host; a virtual cpu is a thread in that process. kill(1), nice(1), top(1) work as expected.

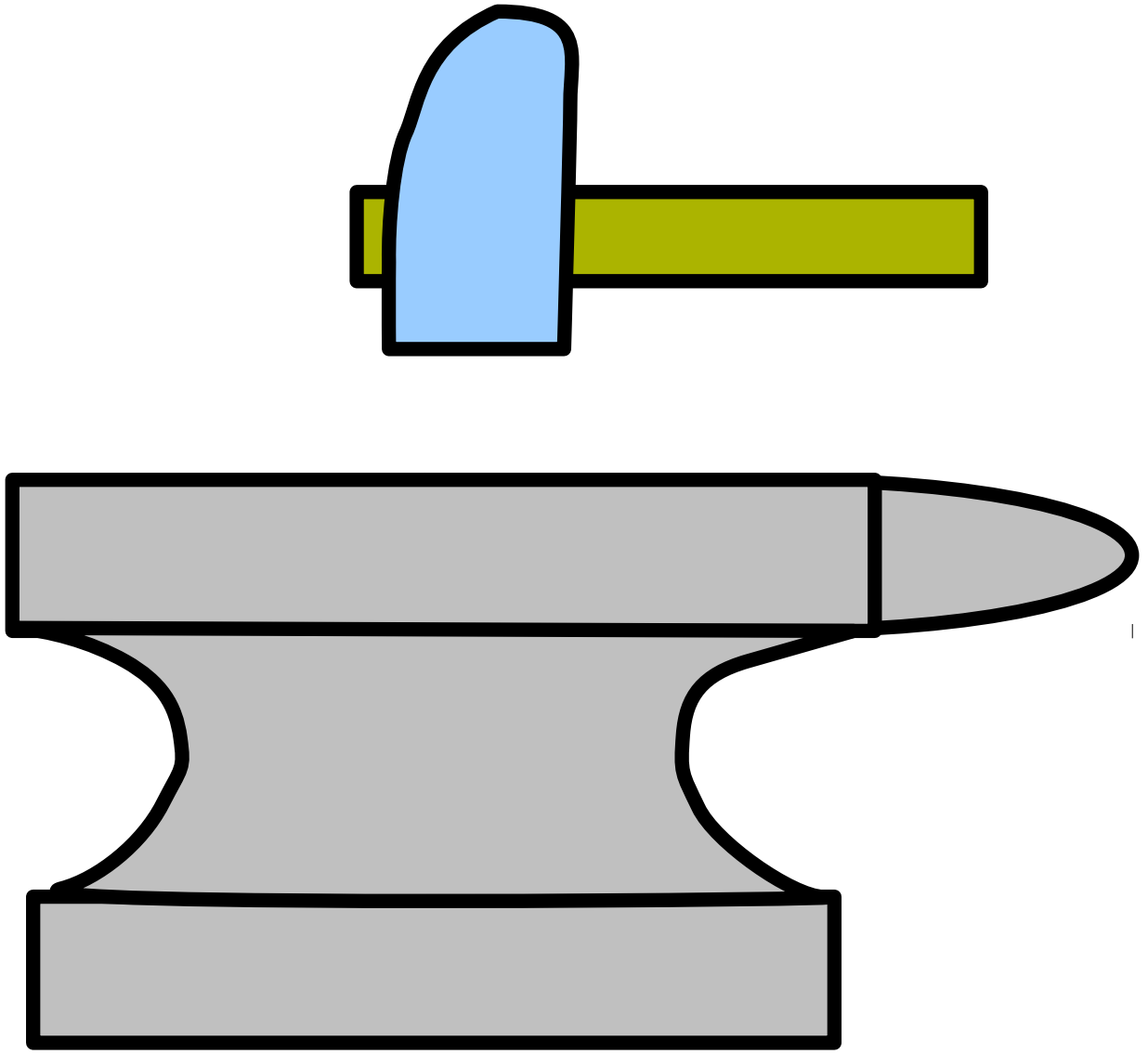
...

--
error compiling committee.c: too many arguments to function

KVM's distinctive features

- Design for future hardware
- Reuse Linux kernel infrastructure
- A witty maintainer

A KVM maintainer's life is a tough one



To integrate or not to integrate?

- Linux community is bigger and can be a bit unfriendly
- But KVM cannot fork the kernel
  - It is not Android!
- Start with quick and painless integration
- Features can get in later
- This was quite successful!

The golden rule of contributing to Linux

Slip your stuff in, in small increments, and with good reasons for why you aren't crazy

Every new crazy feature should be hidden in a nice solid "Trojan Horse" gift: something that looks obviously good at first sight.

-- Linus Torvalds

To integrate or not to integrate?

- Integration with QEMU was much less important
  - x86 KVM usually run with the qemu-kvm fork
  - Several features still implemented only in qemu-kvm
- Three rules for sanity:
  - Work as much as possible with upstream
  - Try to get your stuff upstream
  - Merge regularly
- < 7000 lines of code still to be merged

Crazy features contributed by KVM

- Scheduler notifiers
  - lightweight guest → kernel → guest context switching
- MMU notifiers
  - swapping and overcommitting guest memory
- Samepage merging
- Transparent huge pages

... and for QEMU:

- Virtual machine migration
- SMP guests
- Stable guest hardware
- Virtual machine monitor RPC
- KVM support

KVM's killer feature

- The hypervisor will only run the VM for you
- No policy decision is the hypervisor's business:
  - Security checks
  - Memory management & scheduling
  - NUMA
  - ...
- Not just an “economic” decision, it gives enormous flexibility!

A free software success story

- Built around free components itself
  - Linux, QEMU
  - Successfully contributed back to those projects!
- Stimulated the growth of a large and open community
- Enabled the creation of a large software ecosystem

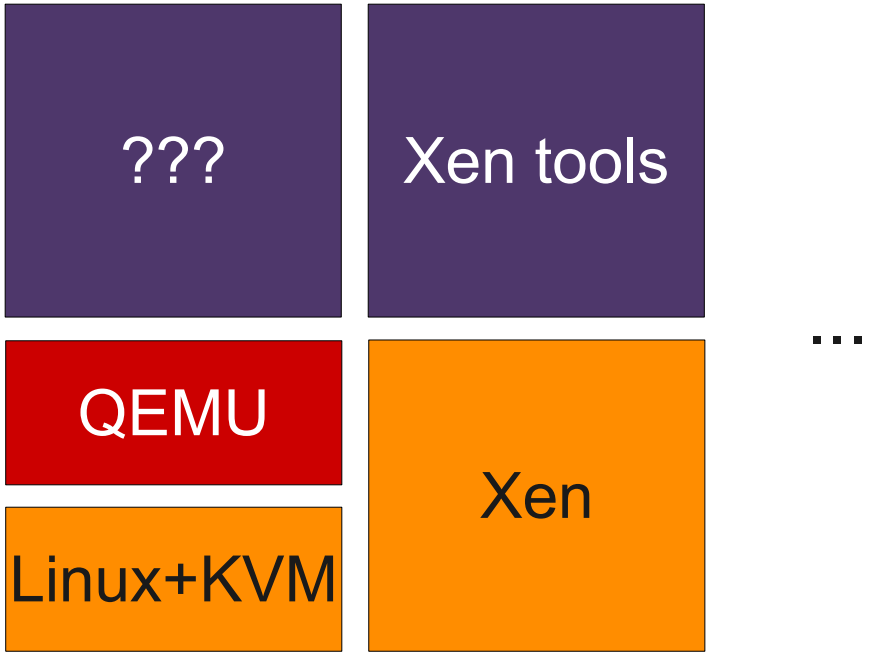
- Technical and social factors reinforce each other!

So, how do you use KVM?

$ qemu-kvm -S -M rhel6.2.0 -cpu Conroe -enable-kvm -m 2048 -smp 1,sockets=1,cores=1,threads=1 -name z-win7x86-1 -uuid e3e19b36-f6b7-4ab9-b604-1f8b5c471bda -smbios type=1,manufacturer=Red Hat,product=RHEL,version=6Server-6.1.0.2.el6_1,serial=50C1C6F0-B18B-11DE-ADF1-00215EC7FC0C_00:1A:64:E7:0E:E0,uuid=e3e19b36-f6b7-4ab9-b604-1f8b5c471bda -nodefconfig -nodefaults -chardev socket,id=charmonitor,path=/var/lib/libvirt/qemu/z-win7x86-1.monitor,server,nowait -mon chardev=charmonitor,id=monitor,mode=control -rtc base=2011-08-04T06:17:36 -boot cdn -device virtio-serial-pci,id=virtio-serial0,max_ports=16,bus=pci.0,addr=0x6 -drive file=/rhev/data-center/6927f974-c6f6-482f-aca9-907c4acc71a9/50027e48-6cb9-4345-9c7a-c22b41ad84d2/images/5ada0ef6-5f4a-40b8-ad92-cb6758de8536/c22f4e68-439b-4a87-8e22-bc7d8e2391f1,if=none,id=drive-ide0-0-0,format=qcow2,serial=b8-ad92-cb6758de8536,cache=none,werror=stop,rerror=stop,aio=native -device ide-drive,bus=ide.0,unit=0,drive=drive-ide0-0-0,id=ide0-0-0 -drive file=/rhev/data-center/6927f974-c6f6-482f-aca9-907c4acc71a9/e0acfcc8-c020-413e-84cd-a93cb0ab9b2d/images/11111111-1111-1111-1111-111111111111/RHEV-toolsSetup_3.0_12.iso,if=none,media=cdrom,id=drive-ide0-1-0,readonly=on,format=raw -device ide-drive,bus=ide.1,unit=0,drive=drive-ide0-1-0,id=ide0-1-0 -drive file=/rhev/data-center/6927f974-c6f6-482f-aca9-907c4acc71a9/50027e48-6cb9-4345-9c7a-c22b41ad84d2/images/f52621e0-8b1e-47af-809c-45de2aa697fc/f77b5dd2-3141-4ea7-84fa-e8cfffe9cff9,if=none,id=drive-virtio-disk0,format=qcow2,serial=af-809c-45de2aa697fc,cache=none,werror=stop,rerror=stop,aio=native -device virtio-blk-pci,bus=pci.0,addr=0x7,drive=drive-virtio-disk0,id=virtio-disk0 -netdev tap,fd=27,id=hostnet0 -device rtl8139,netdev=hostnet0,id=net0,mac=00:1a:4a:23:11:0b,bus=pci.0,addr=0x3 -netdev tap,fd=29,id=hostnet1,vhost=on,vhostfd=30 -device virtio-net-pci,netdev=hostnet1,id=net1,mac=00:1a:4a:23:11:0c,bus=pci.0,addr=0x4 -chardev socket,id=charchannel0,path=/var/lib/libvirt/qemu/channels/z-win7x86-1.com.redhat.rhevm.vdsm,server,nowait -device virtserialport,bus=virtio-serial0.0,nr=1,chardev=charchannel0,id=channel0,name=com.redhat.rhevm.vdsm -chardev spicevmc,id=charchannel1,name=vdagent -device virtserialport,bus=virtio-serial0.0,nr=2,chardev=charchannel1,id=channel1,name=com.redhat.spice.0 -usb -spice port=5902,tls-port=5903,addr=0,x509-dir=/etc/pki/vdsm/libvirt-spice,tls-channel=main,tls-channel=inputs -k en-us -vga qxl -global qxl-vga.vram_size=67108864 -device intel-hda,id=sound0,bus=pci.0,addr=0x5 -device hda-duplex,id=sound0-codec0,bus=sound0.0,cad=0

- Voila! We have a virtual machine
  - Don't forget the fine print

So, how do you use KVM?

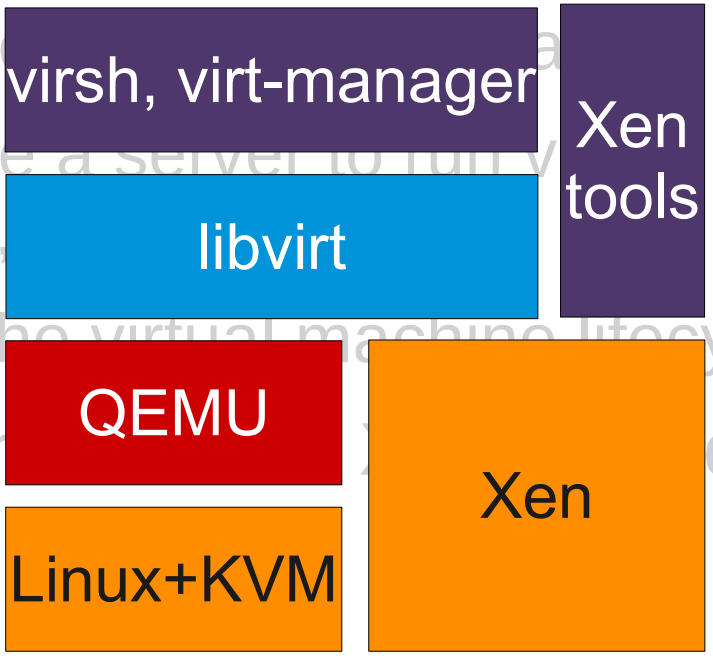


Enter libvirt!

The goal of libvirt: to provide a common and stable layer sufficient to securely manage domains on a node, possibly remote

Enter libvirt!

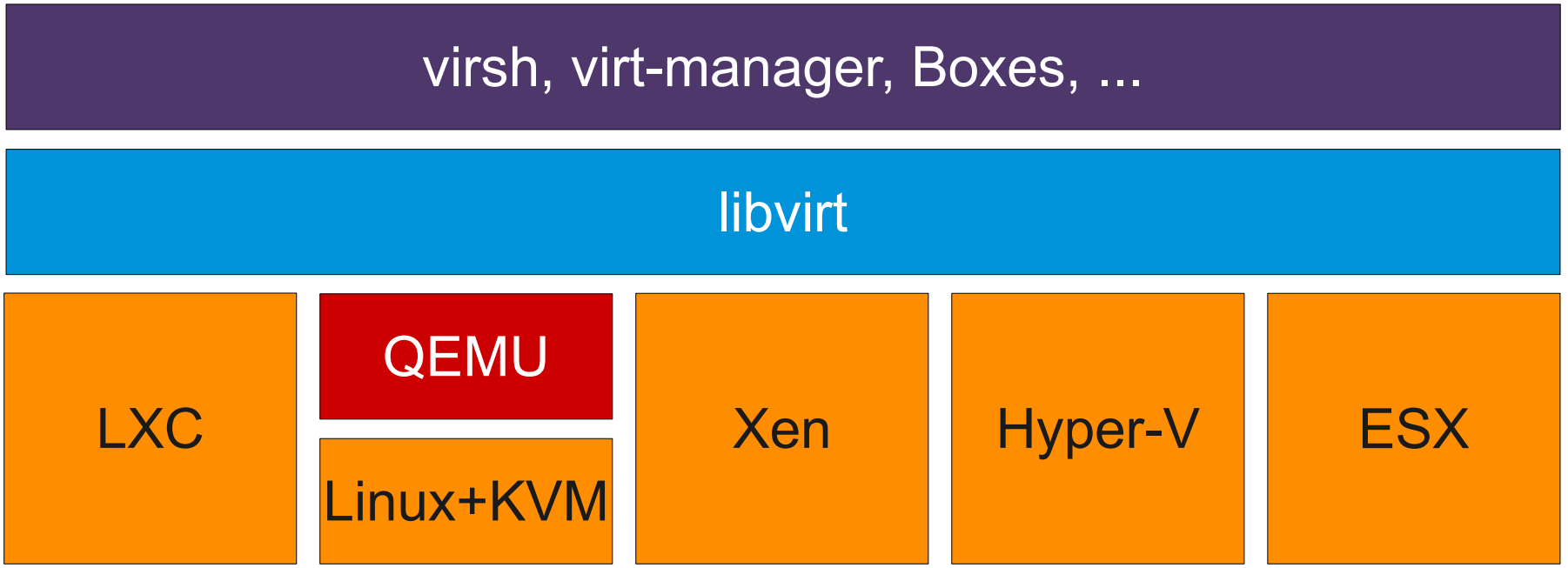
- An API that can be used both locally and remotely
- A daemon to:
  - Handle r
  - Configure a server to run v machine (networking, SELinux,
  - Handle the virtual machine lifecycle for KVM
- Support for d mid 2007



Enter libvirt!

- An API that can be used both locally and remotely
- A daemon to:
  - Handle remote communication
  - Configure a server to run virtual machine (networking, SELinux, ...)
  - Handle the virtual machine lifecycle for KVM
- Support for KVM and Xen around mid-2007
  - Now: Hyper-V, ESX, LXC, UML, ...

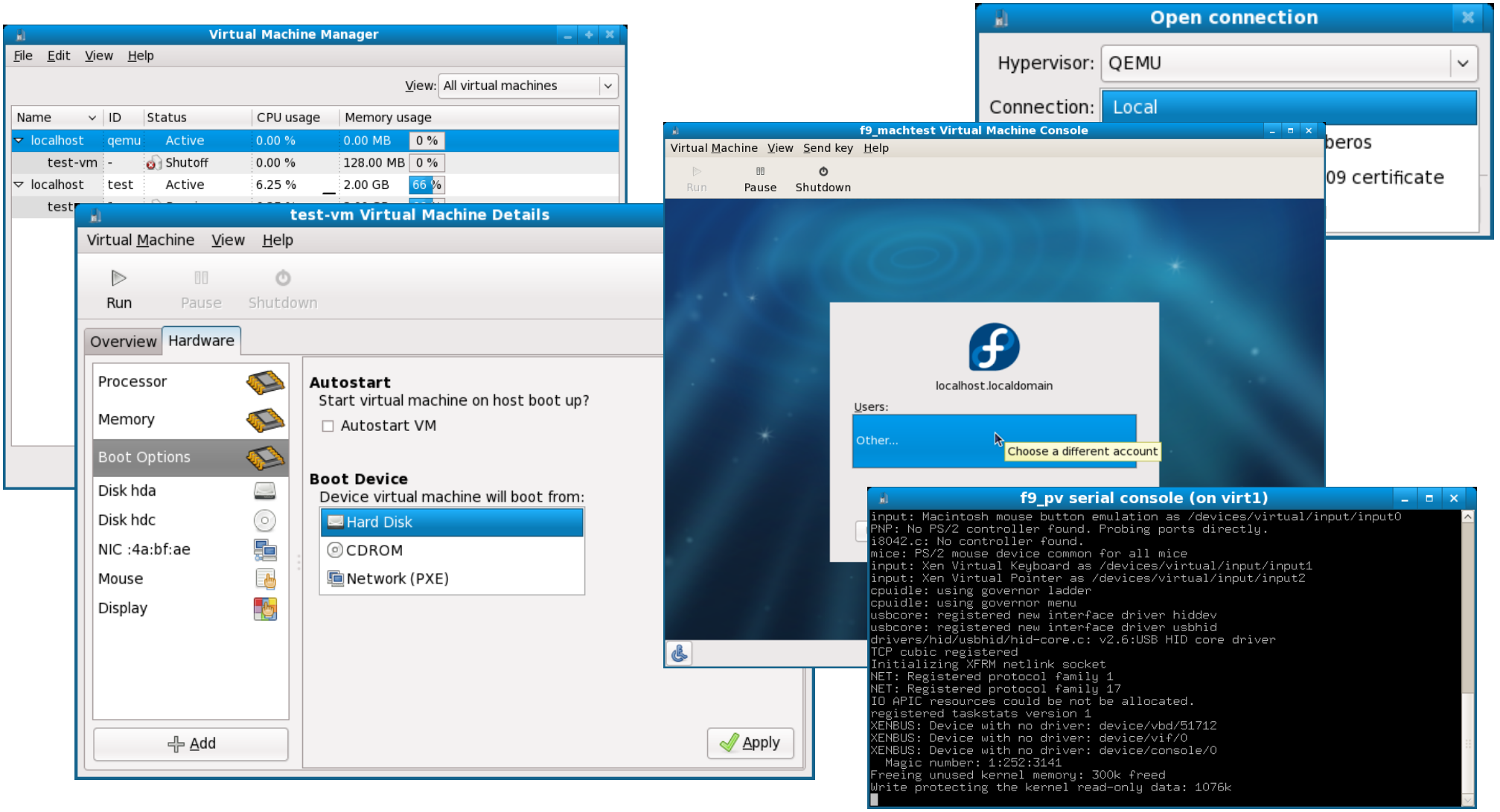
Enter libvirt!



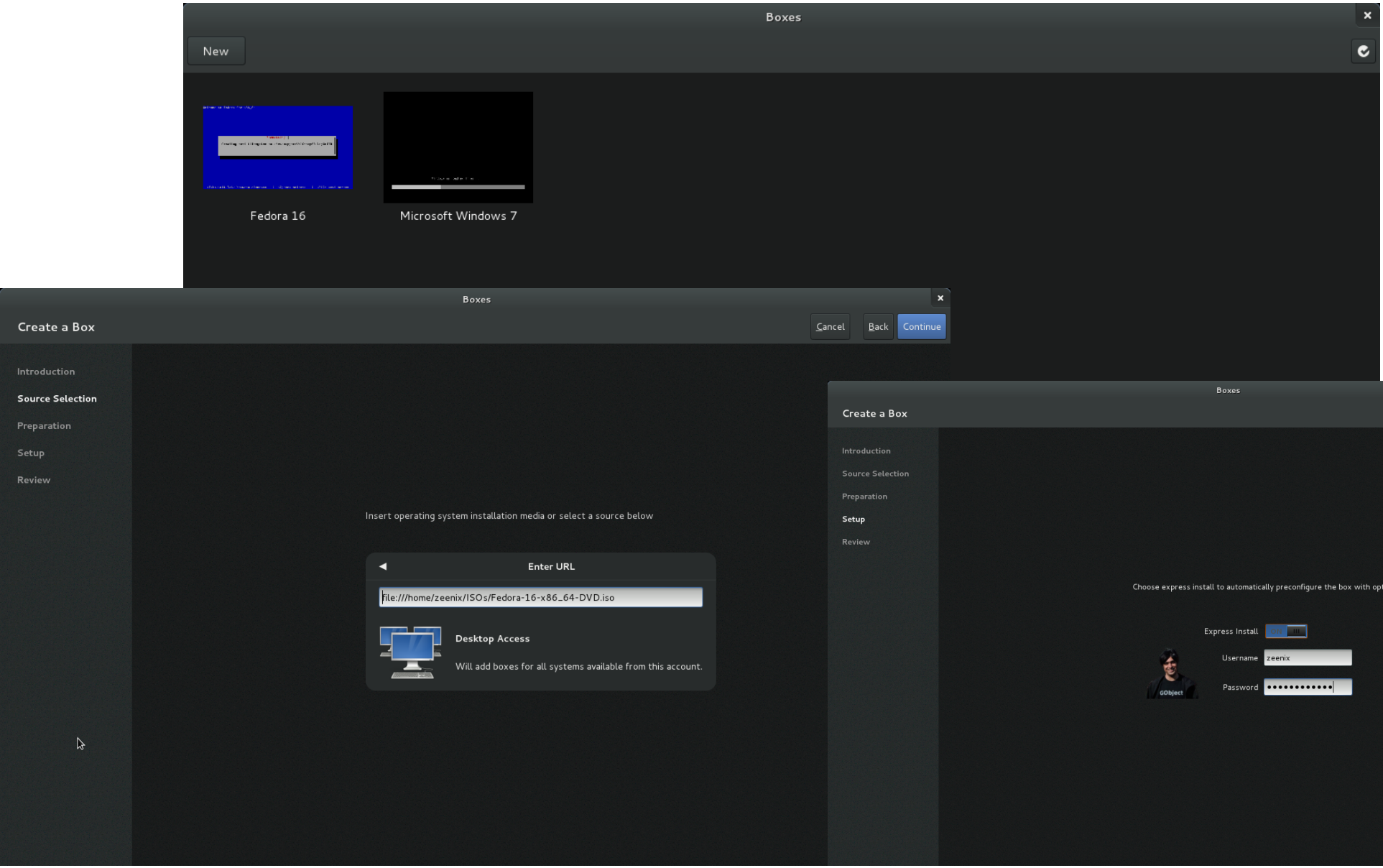
libvirt features

- Comprehensive host management
  - Virtual machines
  - Virtual networks (bridging, NAT, VEPA, ...)
  - Storage (local disks, SAN, NFS)
- Isolation between virtual machines and host
  - Compromised virtual machine cannot access other VMs or hosts
  - Containment in case of hypervisor breaches
  - Based on SELinux and cgroups, shines on KVM!

libvirt users: virt-manager



Libvirt users: Boxes



libvirt users: other APIs

- libvirt-snmp
  - Access domain information via SNMP
- libvirt-qmf
  - Manage hosts, domains, pools via AMQP
  - An agent running within Matahari
- libvirt-cim
  - Open standard to describe guests and resource pools
  - CIM objects abstract the XML schemas

libvirt users: other APIs

- libvirt-glib / libvirt-gobject
  - Access libvirt event loop from GLib
  - Provide a GObject-based API for libvirt objects
  - Usable in many languages via GObject introspection
- libvirt-gconfig
  - an API that abstracts the libvirt XML schemas

Other virtualization tools

- virt-install

  libvirt-based, command-line virtual machine installation

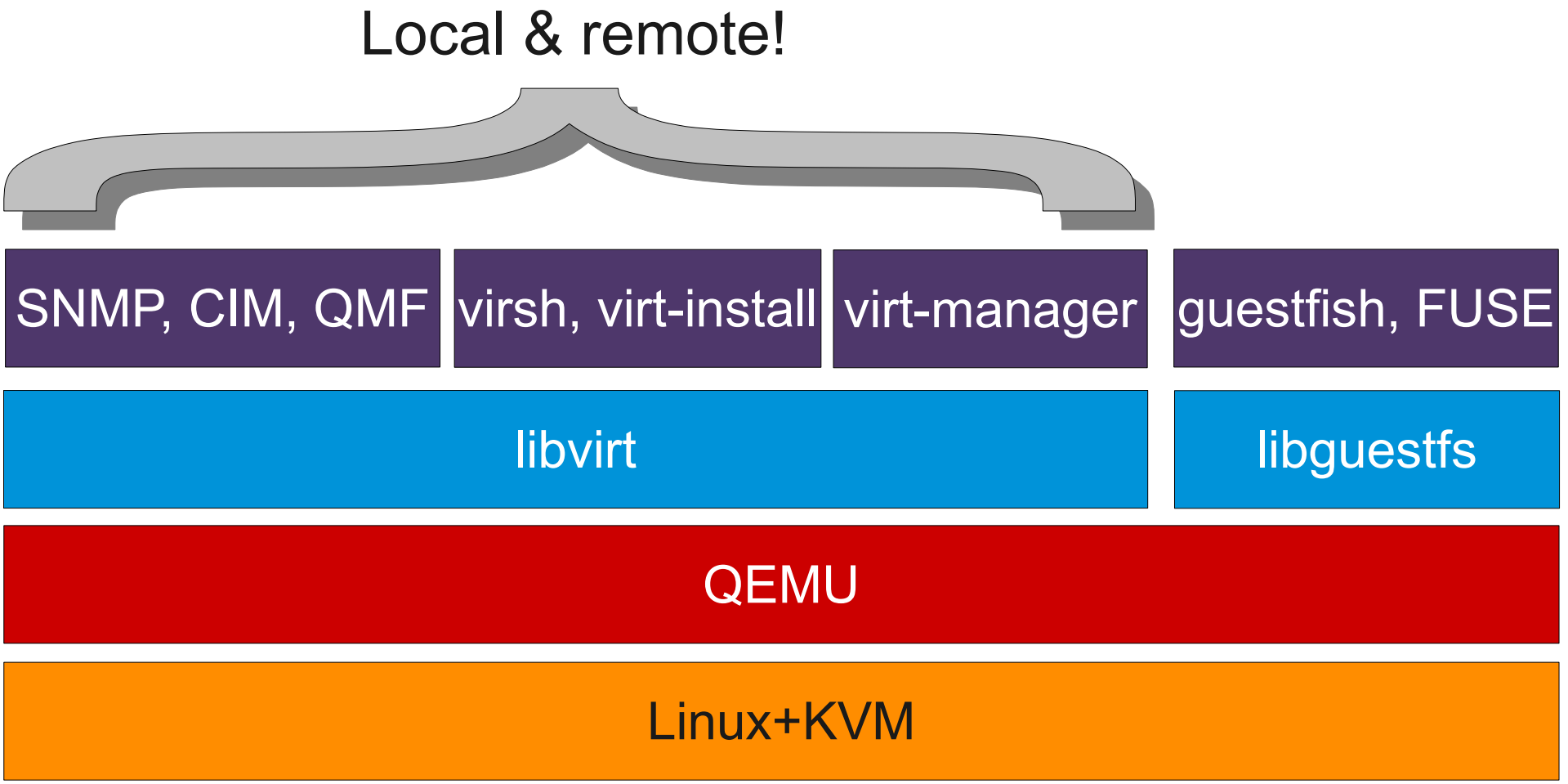
- Oz

  Part of Aeolus project, next-gen virtual machine installer

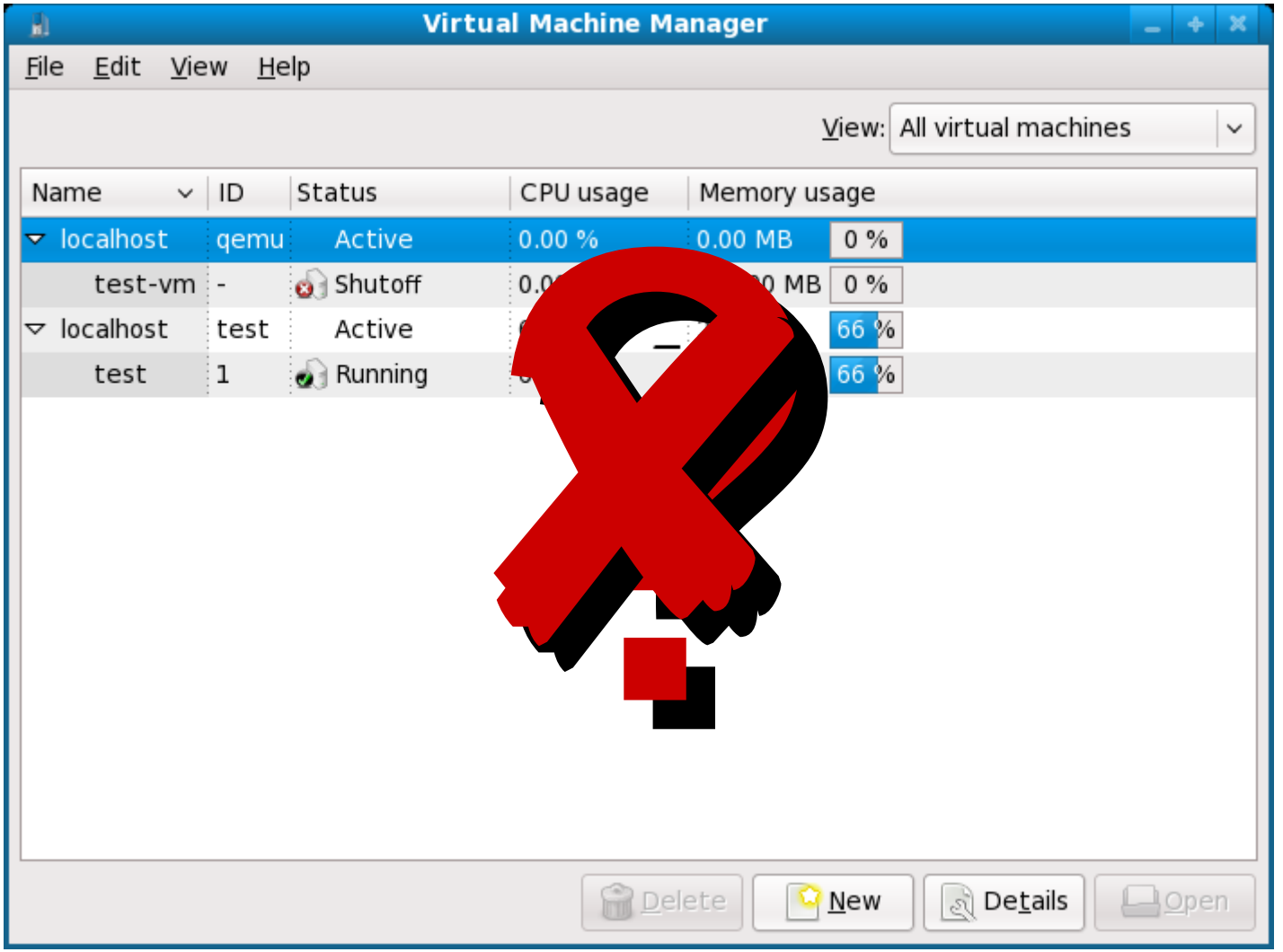
- libguestfs/guestfish

  library and utilities to inspect virtual machine disks

KVM virtualization tools summary



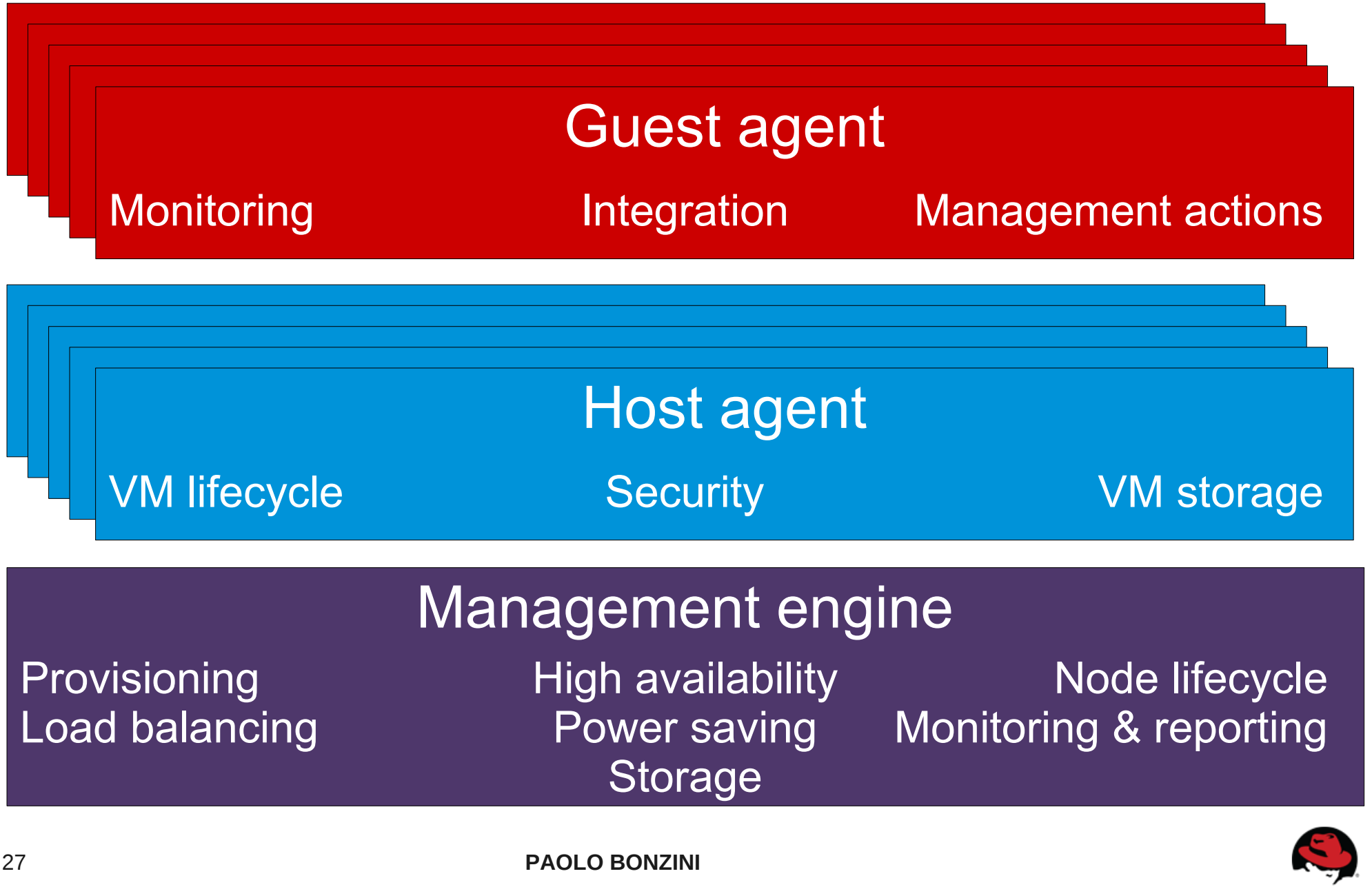
But is this how you really use KVM?



The challenge

- Manage tens of thousands of virtual machines
- Each virtual machine potentially accessible from hundreds (thousands?) of nodes

Large-scale virtualization architecture



Large-scale virtualization solutions

- OpenNebula, OpenStack Compute (Nova)
  - Cloud deployment
- Ganeti
  - Distributed cluster, replicated storage
- oVirt
  - Datacenter virtualization, VDI

Large-scale virtualization solutions

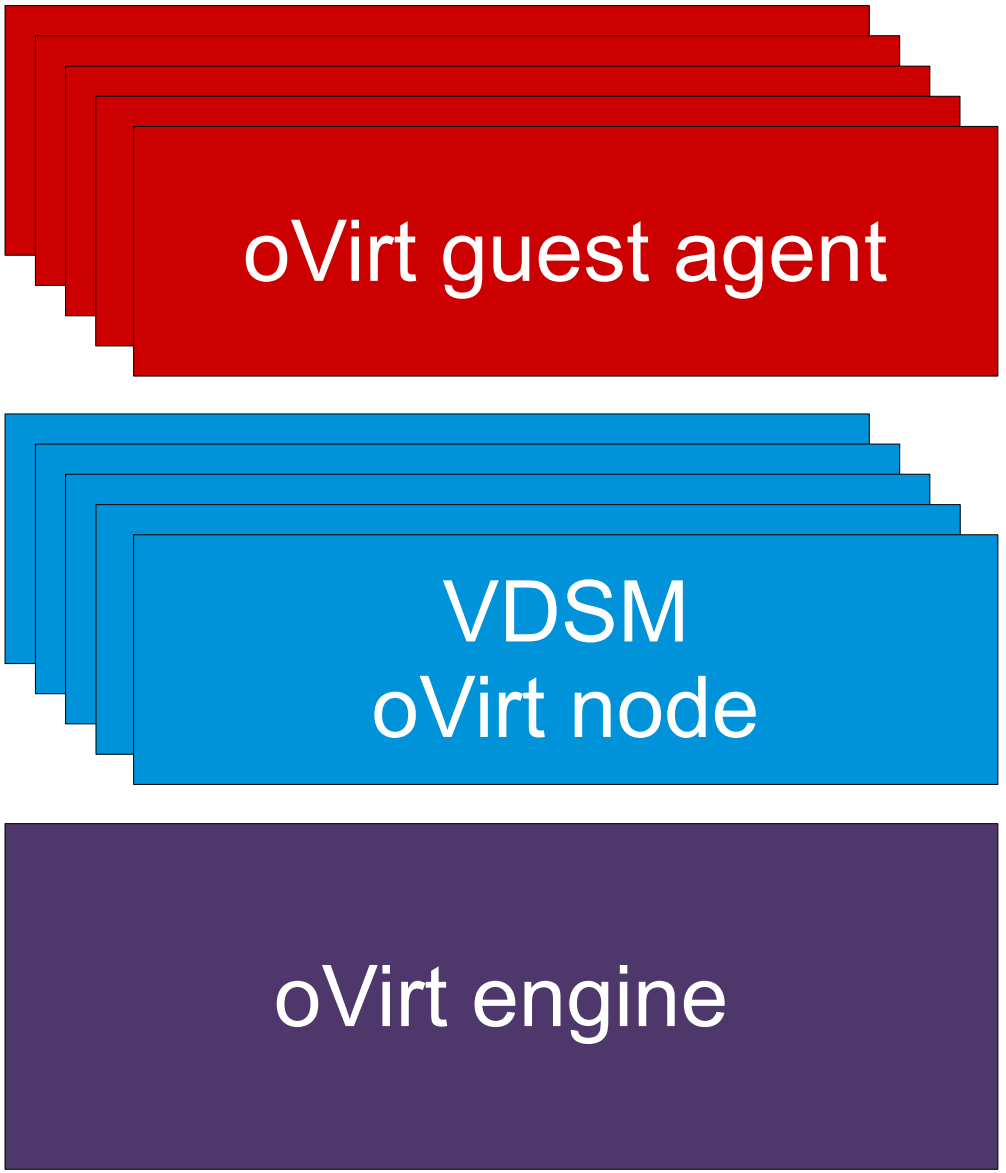
	ONE	Nova	Ganeti	oVirt
Management API	**Yes**	**Yes**	**Yes**	**Yes**
Web control panel	**Yes**	**Yes**	**Yes**	**Yes**
Libvirt-based	**Yes**	**Yes**	No	**Yes**
Live migration	**Yes**	No	**Yes**	**Yes**
Distributed storage	**Yes**	**Yes**	**Yes**	**Yes**
Replicated storage	DRBD	**Ceph**	**DRBD**	No
Specialized distro	No	No	No	**Yes**
Guest agent	No	No	No	**Yes**

Non-KVM hypervisors	**Xen, ESX**	**Many**	**Xen**	None

oVirt

- An open platform for datacenter virtualization
- Comprises multiple integrated projects
- Development also started at Qumranet
- Forms the basis of Red Hat's RHEV 3.0 product
- Backed by Red Hat, IBM, NetApp, Cisco, SuSE, Intel

oVirt



- Single sign-on
- Clipboard management
- Linux + Windows

- Libvirt-based, KVM-only
- Written in Python
- Can run in <100 MB

- REST
- Web portals (admin, user)
- SDK/command line

VDSM

- High-level virtualization management API
- Register with oVirt engine
- VM lifecycle via libvirt
- Storage management (iSCSI, LVM, thin provisioning)
- Monitoring host and VMs
- Interaction with guest agent

Roadmap: oVirt engine

- Complete new web user interface
  - Based on GWT, removes last Windows dependency
  - Integrate reporting into the web UI
- Non-admin API
- Gluster support

Roadmap: oVirt node

- VDSM
  - Live snapshots
  - Live storage migration
  - Shared disks
  - Backup API
  - Service level agreements
- Support additional distributions

Roadmap: libvirt

- Security
  - Fine-grained access control per (user, object, action)
  - Confined LXC containers
- Networking
  - Open vSwitch integration
  - SR-IOV NIC pools
- Interaction with guest agents

Roadmap: QEMU

- Storage
  - virtio-scsi
  - Thin provisioning
  - Live storage migration
- Unified object model
  - Hot-plug improvements

Roadmap: KVM

- Virtualized performance counters
- Networking performance improvements
- Open vSwitch integration

Links and other cool projects

- OpenNebula: http://opennebula.org
- OpenStack: http://openstack.org
- oVirt: http://ovirt.org
- Ganeti: http://code.google.com/p/ganeti/
- libvirt: http://libvirt.org
- Aeolus: http://aeolusproject.org/

- Questions?