Running and Tuning KVM

Hands-on Virtualization

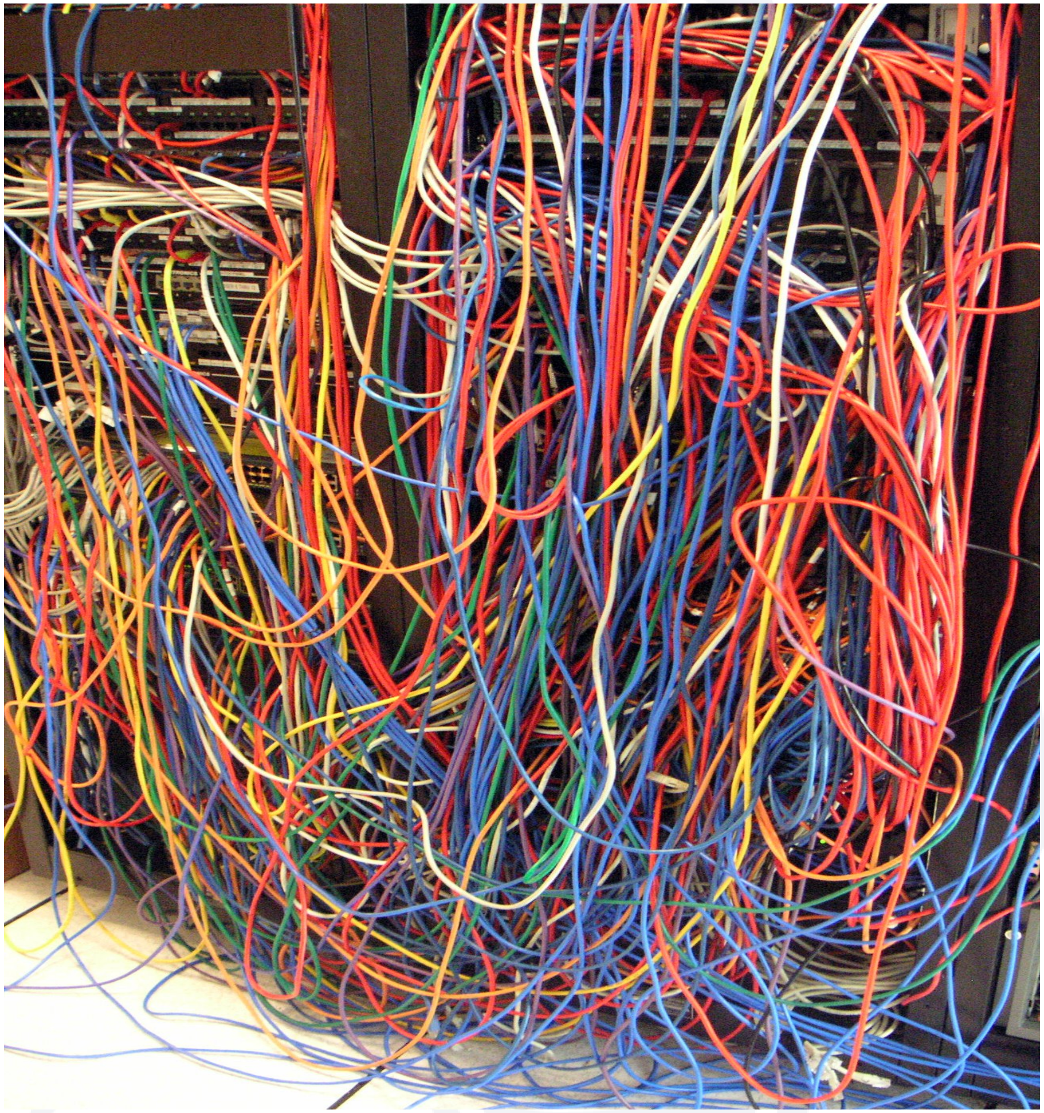
Stefan Hajnoczi – stefanha@linux.vnet.ibm.com
Open Virtualization
IBM Linux Technology Center

2011

Agenda

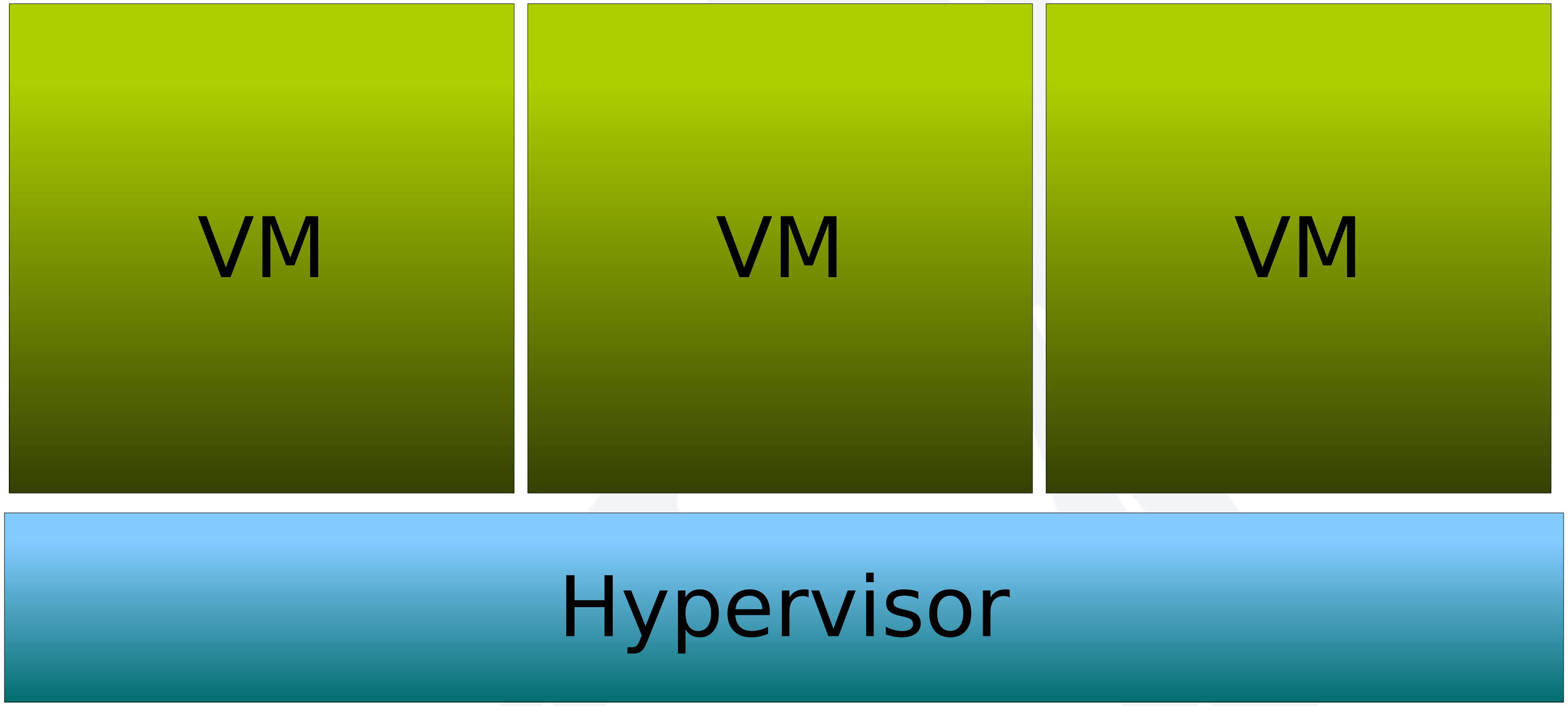
- **Running KVM** - 2:00-3:00 pm
  - Introduction to KVM
  - Hardware & software requirements
  - Installation
  - Creating virtual machines
  - Host configuration
  - Managing virtual machines
- **Advanced KVM -** 3:00-3:50 pm
  - Resource management
  - Performance monitoring
  - APIs and scripting

Server sprawl



Flickr CC by CHRISTOPHER MACSURAK
http://www.flickr.com/photos/macsurak/5020598359/

What is virtualization?



VM

VM

VM

Hypervisor

- Server consolidation
- Development environments
- Running multiple OSes
- Cloud and hosting

What is KVM?

- **Open source**
  - Uses Linux and QEMU
  - Choice of vendor support available
- **Easy to install and maintain**
  - Integrated into the Linux kernel
  - Mainline since 2007 (Linux 2.6.20)
  - Modern distributions already ship it
- **Rich features**
  - Live migration
  - Memory and CPU overcommit
  - Secure remote management
  - Device assignment
  - ...and much more

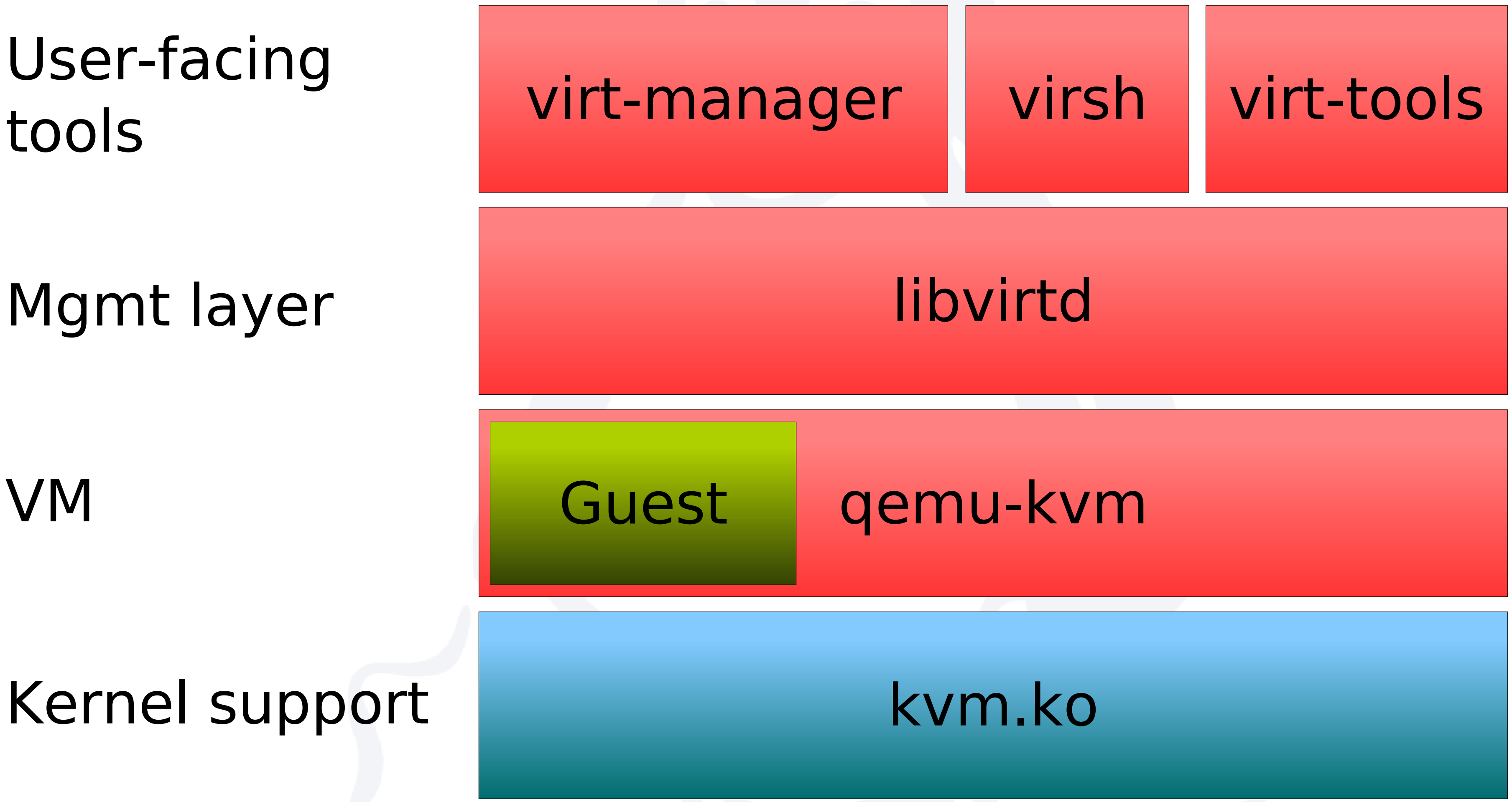
KVM ecosystem

- **KVM as a component**
  - Cloud stacks
  - Embedded/appliances
  - Hide KVM behind their own APIs
- **KVM integration into mangagement tooling**
  - Management and automation stacks
  - Abstract the hypervisor
- **This presentation covers KVM with libvirt**
  - Popular for server virtualization
  - Popular for ad-hoc and desktop
  - Fully open source stack

Layers of the onion

Flickr CC by irish {hunnyb}
http://www.flickr.com/photos/10460483@N02/5448093522/

The KVM stack



User-facing tools

virt-manager

virsh

virt-tools

Mgmt layer

libvirtd

VM

Guest

qemu-kvm

Kernel support

kvm.ko

Hardware requirements

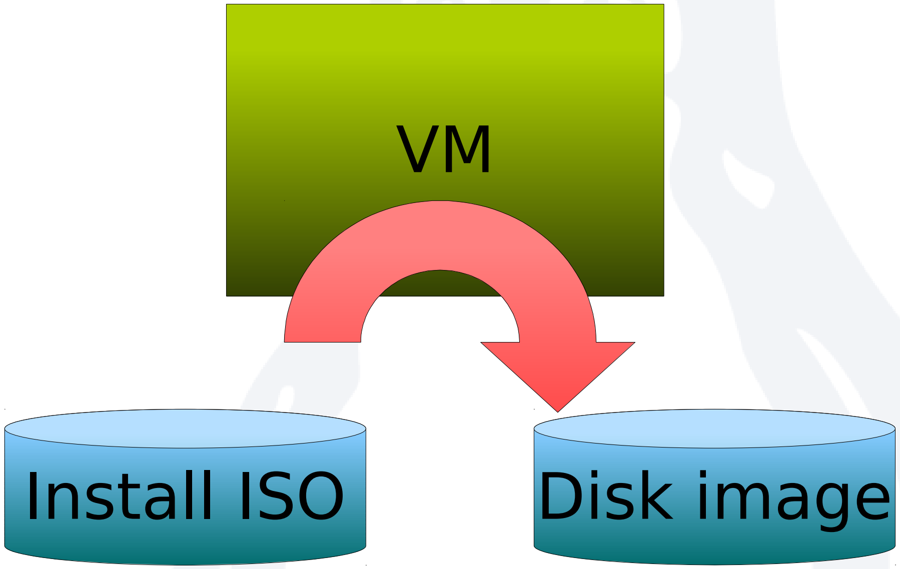
- **32- and 64-bit x86** supported
  - Ports beyond scope of this talk
- CPU with **virtualization extensions**
  - grep '\(vmx\|svm\)' /proc/cpuinfo
- Virtualization extensions can be toggled in BIOS
  - Make sure they are enabled

Software requirements

- **No special kernel components needed**
  - kvm.ko already in kernel package
- **Install userspace components**
  - qemu-kvm – main program
  - libvirt – management layer
  - virt-manager – GUI management tool
- Debian-based distros:
  - aptitude install qemu-kvm libvirt-bin
- Red Hat-based distros:
  - yum install qemu-kvm libvirt
- Add virt-manager if you want a GUI tool

Creating a virtual machine

- VMs can be installed from an ISO file



- Choose size of disk image
  - Small VMs: 10 GB Linux, 16 GB Windows
- Boot VM into installer
  - Kickstart/preseed works like physical install

Options when creating a VM

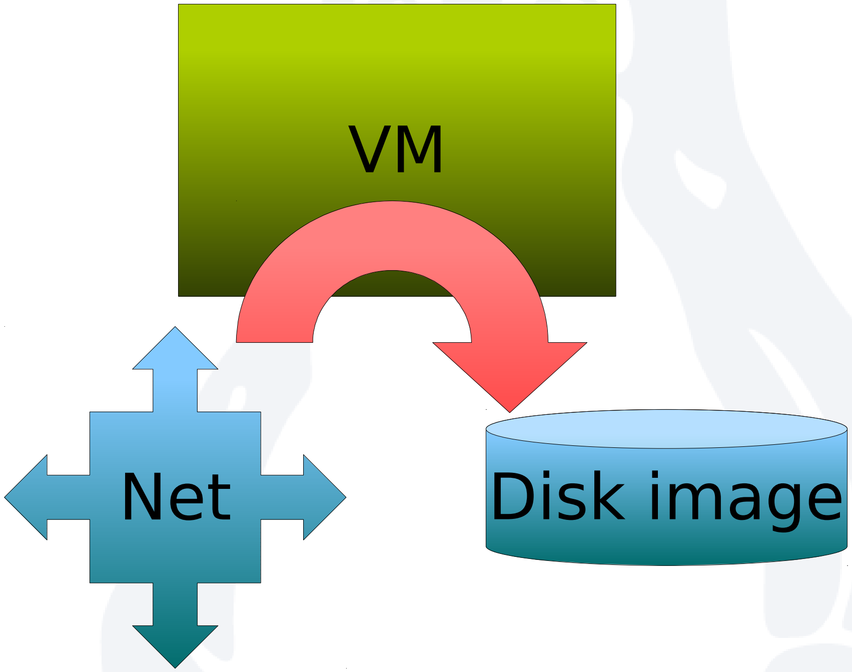
- **Guest OS hint**
  - Older OSes may not support optimizations
  - Automatically chooses good configuration
- **Preallocating disk image**
  - Writing a block for the first time can be slow
  - Allocates entire image ahead of time
  - “Thick provisioning”
- **Virtualization type should be 'kvm'**
- **Choose between 32- and 64-bit VM**
  - 32-bit lowers memory footprint for small VMs

Guest devices (briefly)

- **Networking**
  - Prefer virtio-net for performance
  - Use e1000 for legacy guests
- **Storage**
  - Prefer virtio-blk for performance
  - Use ide for legacy guests
- **USB**
  - USB tablet provides mouse pointer
  - 1000 Hz timer, consider disabling USB

Netboot a virtual machine

- PXE boot allows centralized install over network



- Requires DHCP and TFTP server for install files
- Also useful for stateless VMs

Automated VM install/clone

- **virt-install**
  - Command-line tool to create new VM
  - Both interactive and unattended modes

```
virt-install --name my-vm --ram 1024
             --cdrom fedora15.iso
             --os-variant fedora15
             --file path/to/disk.img
```

- **virt-clone**
  - Command-line tool to copy a VM
  - Sets new UUID, MAC address, etc

```
virt-clone -o rhel-6 --auto-clone
```

Migrating VMs to KVM

- **OSes get upset when hardware changes**
- **Physical-to-virtual migration**
  - Physical server to VM
  - Capture physical disk image
  - Prepare for virtual hardware environment
  - **virt-p2v automates this process**
- **Virtual-to-virtual migration**
  - VMware, Xen, etc to KVM
  - Convert disk image to raw, qcow2, or qed
  - Uninstall old guest tools
  - Prepare for virtual hardware environment
  - **virt-v2v automates this process**

Flickr CC by signal11
http://www.flickr.com/photos/signal11/722495325/

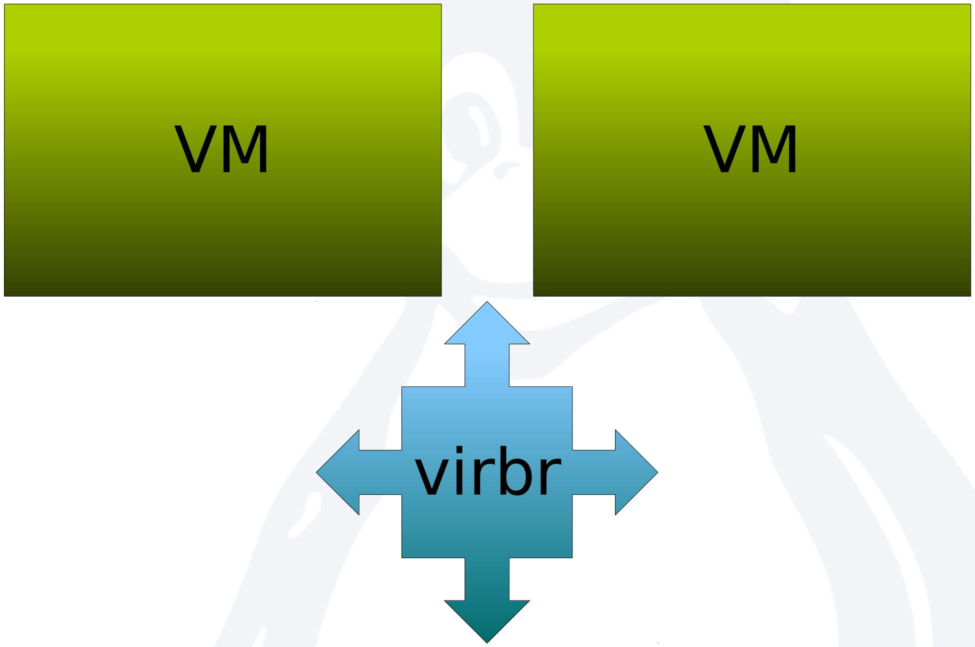
Host configuration

- Hypervisor, storage, and network configuration
  - Host-wide settings
- KVM scales to a range of use cases
  - Think Linux on phones to supercomputers
  - There are many possible configurations

Networks

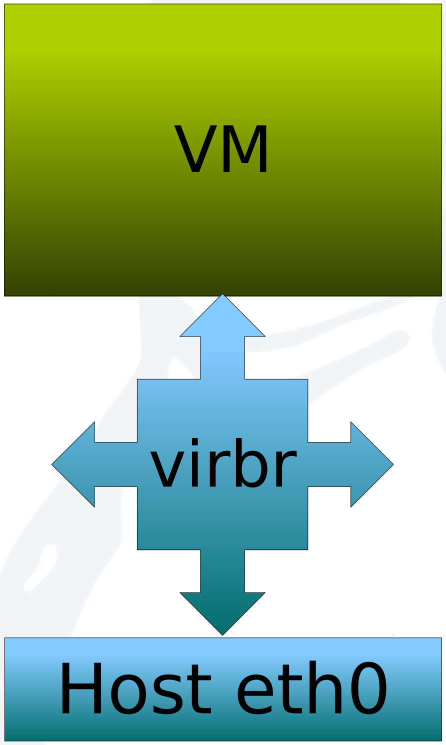
- **VMs have virtual ethernet adapters**
  - Libvirt can set up physical net connectivity
- **Services available for VM networking**
  - DHCP
  - Firewall rules
  - Traffic shaping (QoS)
- **Several net configurations are support**
  - Depending on use-case
  - Let's look at them in turn

Private network



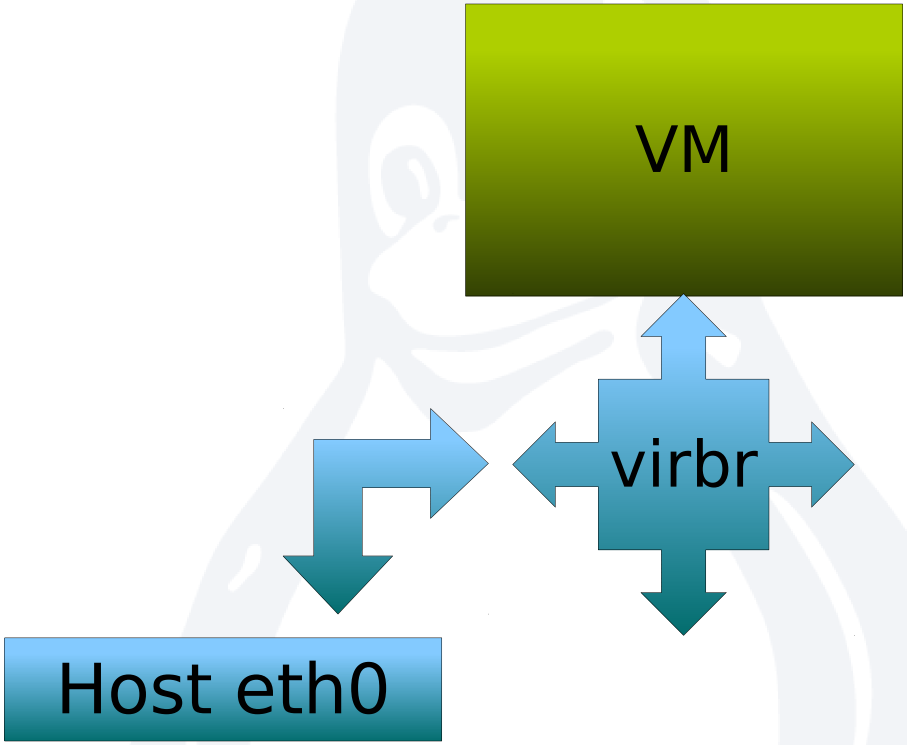
- **No physical network connectivity**
  - Guarantees VMs are isolated
- **VMs can communicate with each other**
  - Or be totally isolated with independent private networks

Bridged network



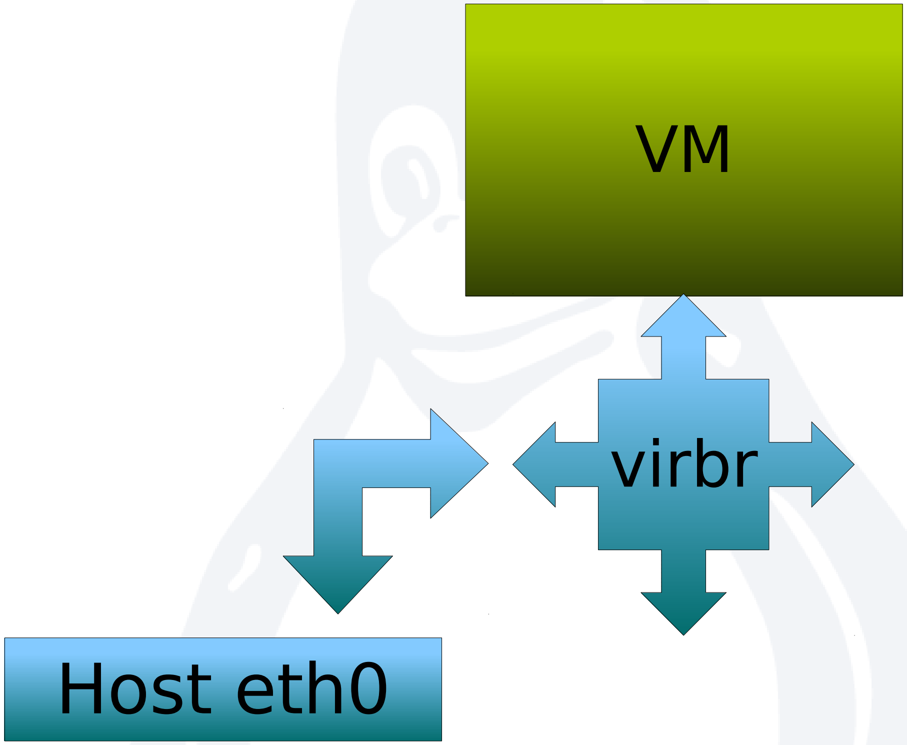
- **VM is visible on host network**
  - Needs host network DHCP or static IP
- **VM appears like physical machine**
  - Suitable on managed LAN and datacenter
  - Not suitable on foreign LAN

Routed network



- **Host acts as router for VM**
  - Host has full firewall and routing capability
- **Popular on managed LAN**
  - LAN must know to route VM traffic to host

NAT network



- **Host acts as a masquerading router**
  - VM does not need IP on host network
  - Cannot connect back to VM by default
- **Popular on laptops and desktops**
  - Usable on foreign networks

Storage pools

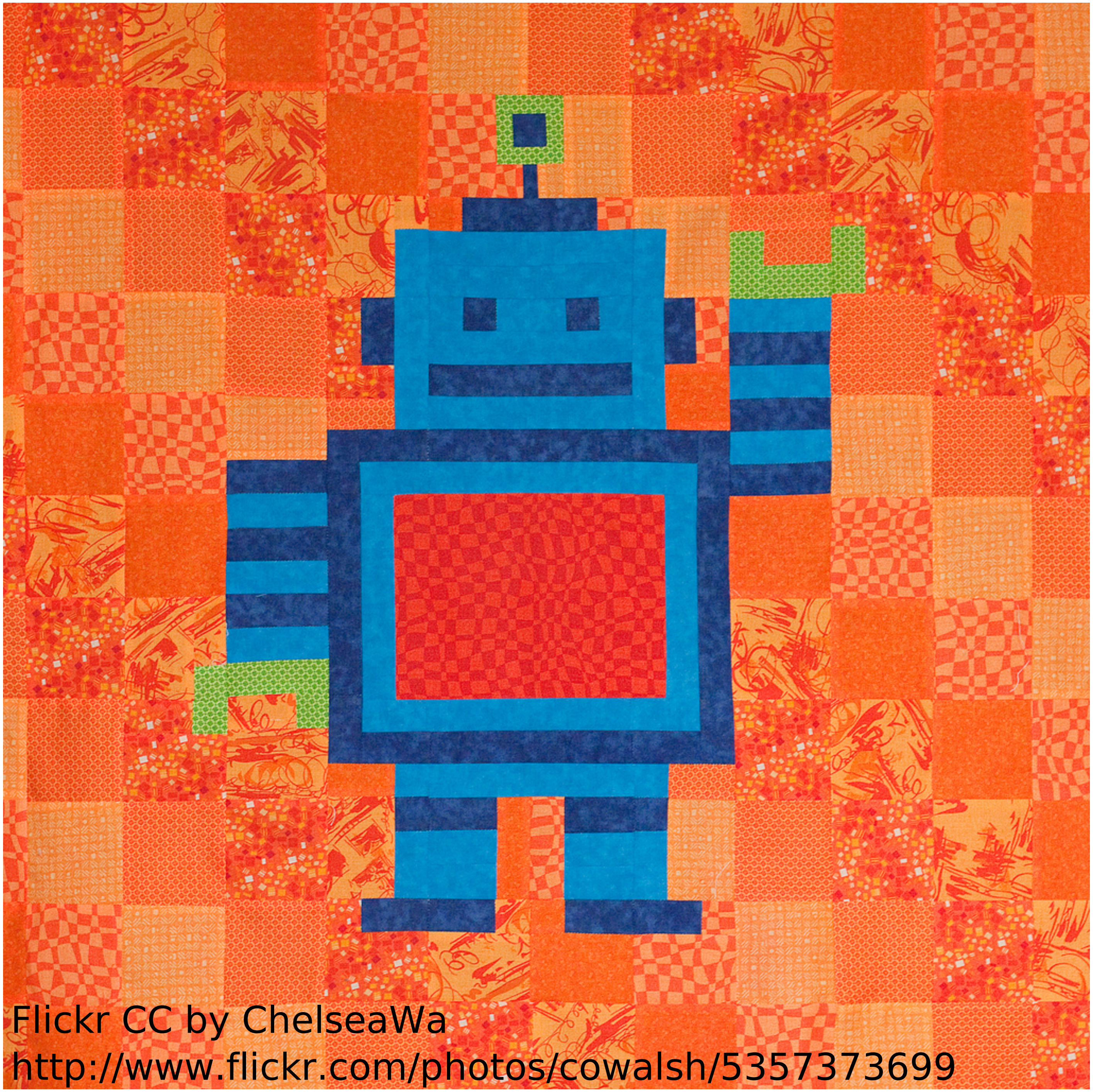
- **Virtual disks come from a storage pool**
  - Libvirt calls virtual disks “storage volumes”
  - Commands to create, delete, etc volumes
- **Storage pools represent storage backends**
  - Local directories on a host file system
  - Local LVM volumes
  - Remote directories over NFS
  - Remote iSCSI LUNs
- **Ad-hoc disk images can also be configured**
  - Useful for managing storage outside libvirt
  - Just specify path to image file or block device

Local storage

- **LVM volume group**
  - Each virtual disk is an LVM volume
  - Familiar management tools
  - Good performance due to thin layer
- **Directory on host file system**
  - Each virtual disk is a local file
  - Can use raw files or image format
  - Raw performance fairly good
  - Image formats add features at a cost
    - Compact – efficient transfer (e.g. HTTP)
    - Backing files – clone from master image

Remote storage

- **Directory over NFS**
  - Centralized storage, easy management
  - Efficient live migration between hosts
  - Complex performance characteristics
    - More layers and network factors
- **iSCSI or FC LUNs**
  - Remote block storage
  - Good fit for traditional enterprise setups
  - Can also put LVM onto remote LUN for local management



Flickr CC by ChelseaWa
http://www.flickr.com/photos/cowalsh/5357373699

Connecting to VM screen

- **KVM uses VNC**
  - Next-gen Spice protocol under development
- **virt-viewer -c <uri> <domain>**
  - Local:
  - virt-viewer -c qemu:///system vm1
  - Remote:
  - virt-viewer -c qemu+ssh://host vm1
- **Regular VNC client works too**
  - First find VNC display number:
  - virsh vncdisplay vm1
  - Then connect:
  - vncviewer host:$displaynum

virsh command-line interface

- **Provides commands for libvirt APIs**
  - virsh shutdown vm1
- **More low-level than virt-manager**
- **VM config expressed in “domain XML”**

```
<domain type='kvm'>
   <name>vm1</name>
   <memory>1048576</memory>
   <vcpu>1</vcpu>
...
```

- **Networks and storage pools also in XML**
- **Host-wide configuration in /etc/libvirt**

Flickr CC by qwrrty
http://www.flickr.com/photos/qwrrty/5877478960/

Resource management

- Hosts resources:
  - CPUs
  - Memory
  - I/O bandwidth and iops
- Resource management questions:
  - How do I dedicate a CPU to this VM?
  - Can I run multiple VMs on the same CPU?
  - How do I add memory to a running VM?
  - Can I give VMs more memory than the host?

CPU resource management

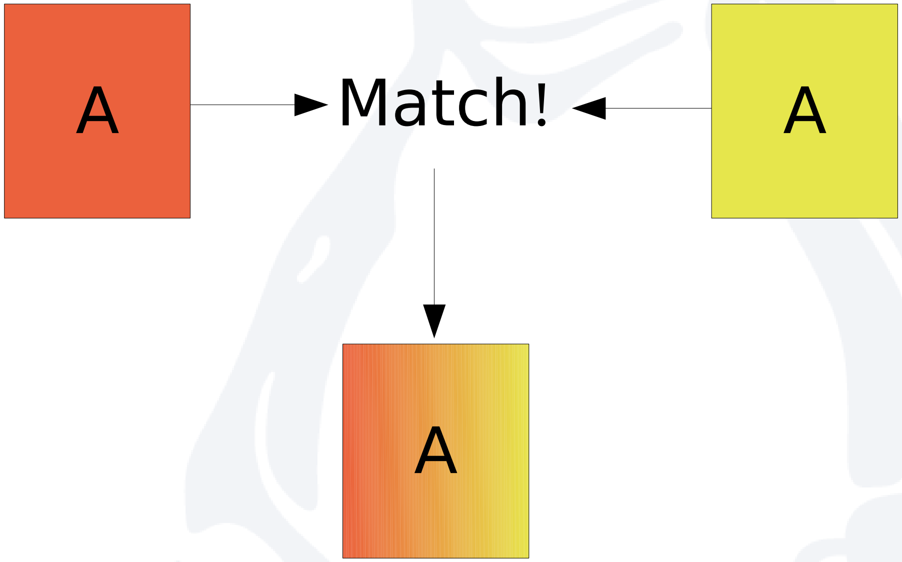
- Each vCPU is a thread on the host
- Normal Linux thread scheduling applies:
  - Multiple threads run on a host CPU by default
- Running more vCPUs than host CPUs
  - Expect low performance
  - Avoid SMP guests when overcommitting
- Use affinity to **bind vCPU**:
  - virsh vcpupin <dom> --vcpu 0 1
  - Pins vCPU 0 to host CPU 1
- Use cgroup cpuacct controller for **CPU share**:
  - virsh schedinfo <dom> --set cpu_shares=X
  - Sets proportional CPU share

Memory resource managment

- **Adjusting memory of running VM**
  - virsh setmem <dom> <kilobytes>
  - Must be within max memory value
  - Set max memory when booting VM
- **Memory can be overcommitted**
  - Simply assign more memory than available
  - Host will swap in guest memory
  - Test performance before deploying

Kernel Samepage Merging

- Shares identical memory pages between VMs
  - Reduces memory consumption on host
  - Ideal when running cloned VMs



- Enable with: echo 1 >/sys/kernel/mm/ksm/run
- Consumes CPU to find matching pages
  - Use ksm when sharing is likely

Flickr CC by Dawn Hopkins
http://www.flickr.com/photos/seenoevil/343753843/

Performance monitoring

- **VM performance is different**
  - Can have different OSes
  - Might be owned by someone else
  - Host cannot see inside VMs
- **Performance questions**
  - Why is my VM slow?
  - Will I need to upgrade the host's RAM?
  - Which VM is hogging the disk?

Host-wide monitoring

- qemu-kvm is just a userspace process
  - VM executes as part of qemu-kvm
  - Standard performance tools can be used
- vmstat 1

```
procs -----------memory---------- ---swap-- -----io----
 r  b   swpd   free   buff  cache   si   so    bi    bo
 0  0  64184 224432 226104 1627340    0    0     0     0
-system-- ----cpu----
in   cs us sy id wa
5191 6389 19  4 78  0
```

- System load (r – runnable, b – blocked)
- Memory utilization
- Swap activity (si – swap in, so – swap out)

Host-wide monitoring (cont'd)

- Disk I/O (bi – blocks in, bo – blocks out)
- CPU utilization (us – user, sy – system, id – idle, wa – I/O wait)

- This is standard Linux performance monitoring
  - Works because qemu-kvm is a process
- Let's look at answering specific questions...

Host-wide CPU utilization

- **How much CPU utilization is inside VMs?**
- On host: mpstat -A 1

```
CPU    %usr   %nice    %sys %iowait    %irq   %soft
All     1.50    0.00    1.50    0.00    0.00    0.00
0       2.97    0.00    1.98    0.00    0.00    0.00
1       1.98    0.00    1.98    0.00    0.00    0.00
%steal  %guest   %idle
0.00    1.00   96.00
0.00    0.00   95.05
0.00    0.99   95.05
```

- Useful since %guest not displayed by vmstat

Per-VM CPU utilization

- **Which VM is consuming CPU?**
- top -c
    - Interactive list of processes
    - Look for top qemu-kvm process
- pidstat -ul -C kvm 1
    - Only shows kvm processes
    - See how much each VM is consuming

Host memory utilization

- **Is the host running low on physical RAM?**
- Check current host memory utilization
  - free -m

```
              total      used      free    shared   buffers    cached
Mem:           3862      3656       206         0       226      1596
-/+ buffers/cache:       1834      2028
Swap:          1903        60      1843
```

- Watch for swap activity
  - vmstat 1
  - Check swap in/out ('si'/'so') counts
- Remember guest memory can be swapped out
  - Overcommit is possible, plan accordingly

Host-wide disk I/O

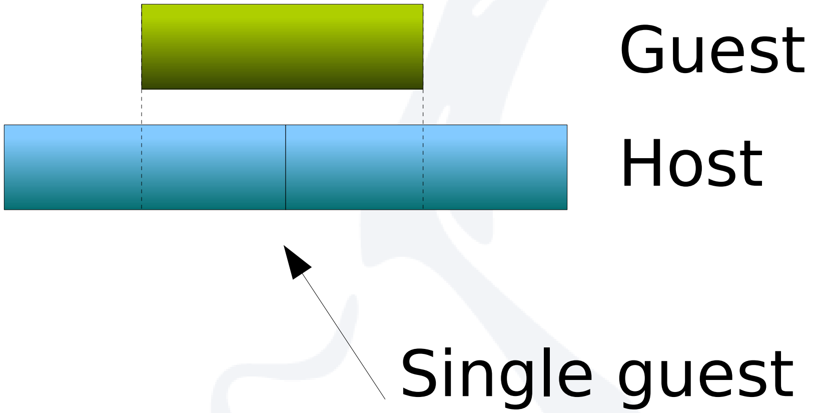
- **What type of I/O is the host performing?**
- iostat -k -x

```
Device:         rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s
dm-3              0.00     0.00    0.00 1179.00     0.00 20604.00
avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
34.95     0.86    0.73    0.00    0.73   0.73  85.60
```

- You must know which host block device
  - /var/lib/libvirt/images → /dev/mapper/root
  - Use mount to find host block device
  - Use dmsetup table to find physical device

Host-wide disk I/O (cont'd)

- If you have access to VM
  - Compare I/O pattern on host to guest
  - I/O pattern should be very similar
  - Average request size (avgrq-sz)
  - Significant difference could mean misalignment

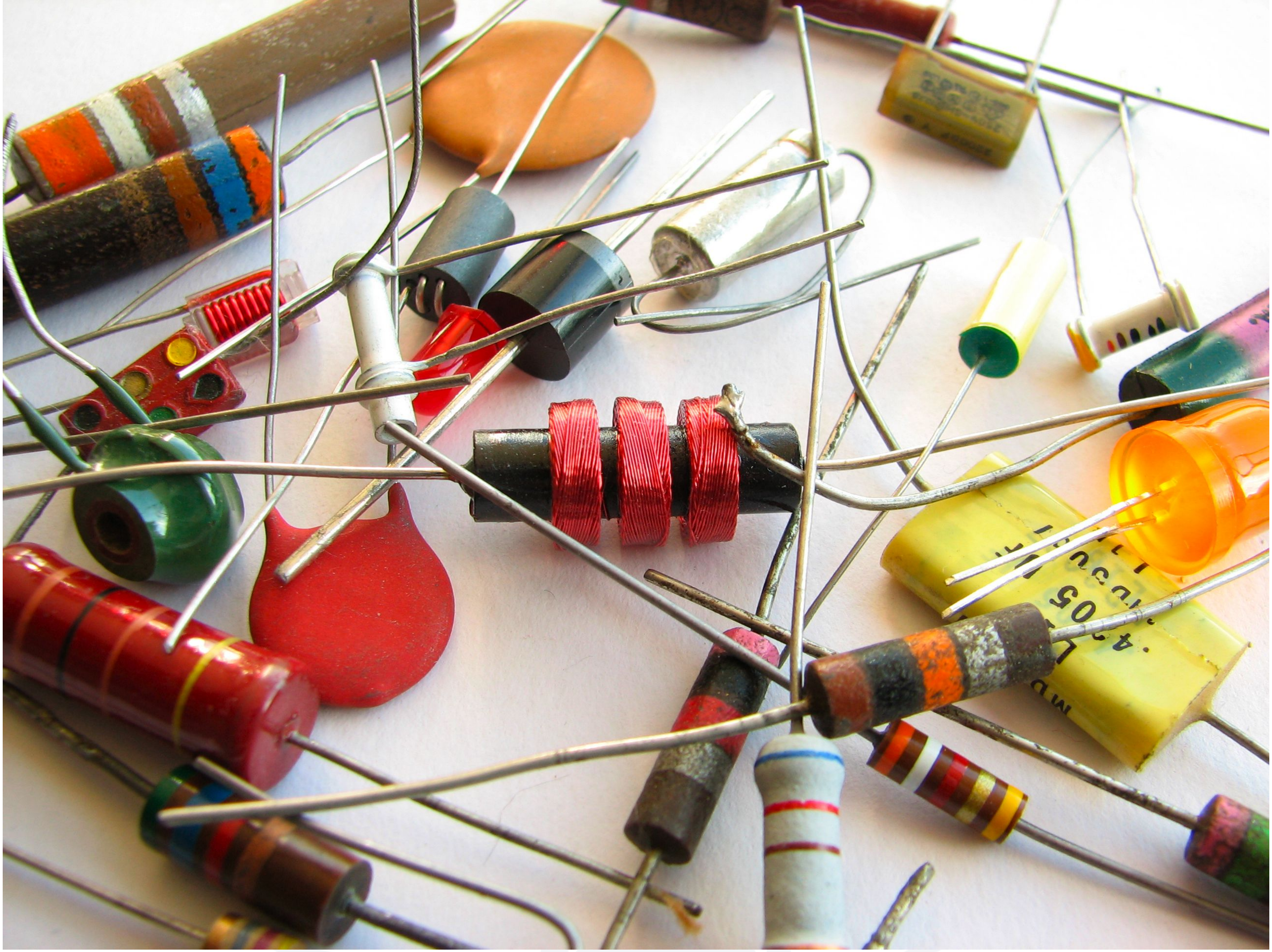


Single guest block spans host blocks, requires accessing two blocks instead of one.

Per-VM disk I/O

- **Which VM is hogging the disk?**
- Check qemu-kvm processes doing heavy I/O
  - pidstat -dl -C kvm 1

```
PID   kB_rd/s   kB_wr/s kB_ccwr/s  Command
9291 8992.00     24.00      0.00  /usr/bin/kvm
```



Flickr CC by Windell Oskay, http://www.flickr.com/photos/oskay/437341128/

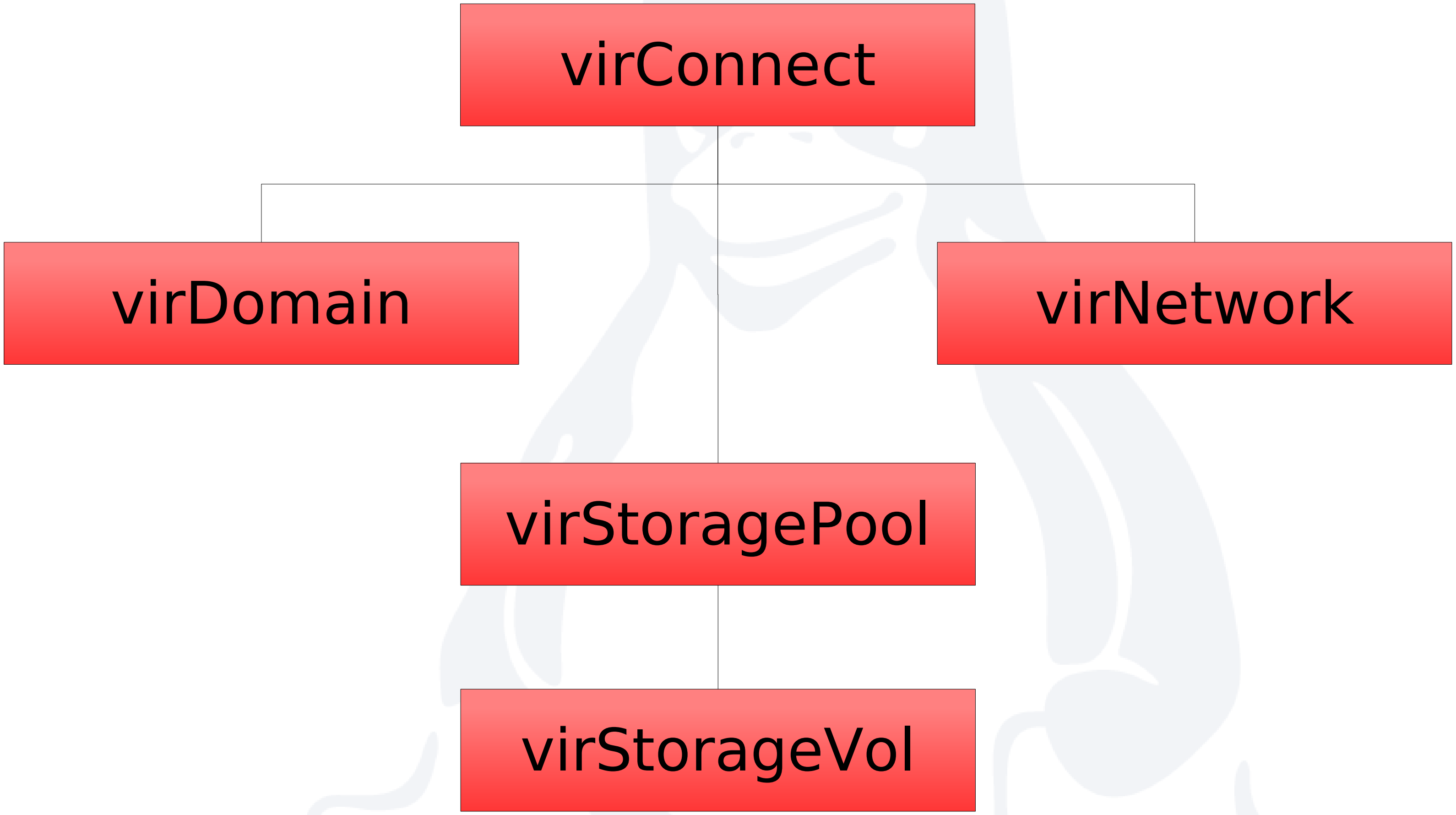
APIs and scripting

- Libvirt provides APIs and command-line tools
- Bindings
  - C/C++, Python, Perl, Java, OCaml, ...
- Areas covered
  - VM lifecycle
  - Monitoring and connecting to VMs
  - Storage pools
  - Networking and firewall rules
- http://libvirt.org/

virsh – command-line tool

- Basic scripts can use virsh
  - Especially useful for shell scripting
  - Easy interactive mode for development
- Points to consider
  - Interface not guaranteed stable
  - More efficient to use API

Libvirt API



Main types in the API

Python example

- Halve assigned memory:

```
>>> import libvirt
>>> c = libvirt.open("qemu:///system")
>>> dom = c.lookupByName('vm1')
>>> dom.maxMemory()
1048576
>>> dom.setMemory(1048576 / 2)
0
```

More libvirt APIs

- Official C API documentation:
  - http://libvirt.org/html/libvirt-libvirt.html
- Existing virt-tools can serve as examples:
  - C – virsh
  - Python – virt-clone, virt-install, virt-manager
  - OCaml – some of libguestfs

Where to go for more info

- QEMU, KVM, and libvirt
  - http://qemu.org/
  - http://linux-kvm.org/
  - http://libvirt.org/
- virt-tools – management tools
  - http://virt-tools.org/
- libguestfs – manipulating disk images
  - http://libguestfs.org/
- My blog – poweruser and developer tips
  - http://blog.vmsplice.net/

Stay current on Linux and Open Virtualization at IBM

Linux

Follow us on Twitter: @Linux_at_IBM

Like us on Facebook: Linux at IBM

www.ibm.com/linux

Linux At IBM

@linux_at_ibm

Linux is at the forefront of smarter solutions. IBM provides complete Linux solutions: top-to-bottom, end-to-end.

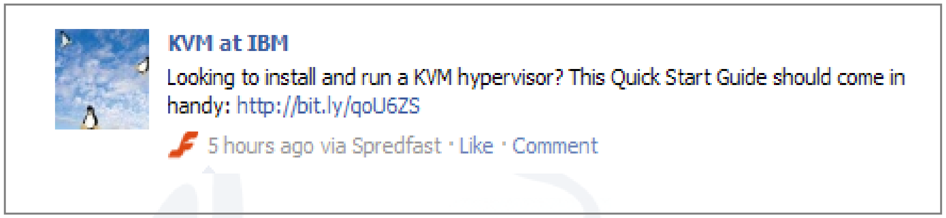
http://www.ibm.com/linux/

Open Virtualization & KVM

Follow us on Twitter: @OpenKVM

Like us on Facebook: KVM at IBM

www.ibm.com/systems/kvm



KVM at IBM

Looking to install and run a KVM hypervisor? This Quick Start Guide should come in handy: http://bit.ly/qoU6ZS

5 hours ago via Spredfast · Like · Comment