Libvirt

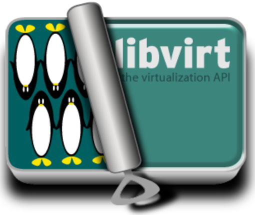
Hypervisor Independent Virtual Machine Management

Guido Günther *<agx@sigxcpu.org>*

2009-07-28

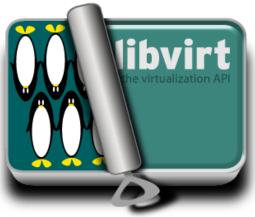
What is it

- C Library for managing virtual machines
- Why's that useful:
  - lots of different hypervisors in Debian: QEMU, KVM, Xen, VirtualBox, LXC, OpenVZ, UML
  - some of these change command line syntax and ABI every minute



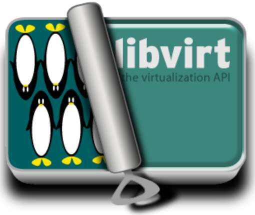
What is it

- C Library for managing virtual machines
- Why's that useful:
  - lots of different hypervisors in Debian: QEMU, KVM, Xen, VirtualBox, LXC, OpenVZ, UML
  - some of these change command line syntax and ABI every minute
  - so libvirt aims to provide
    - stable command interface
    - stable configuraton format
    - stable API



What is it

- C Library for managing virtual machines
- Why's that useful:
  - lots of different hypervisors in Debian: QEMU, KVM, Xen, VirtualBox, LXC, OpenVZ, UML
  - some of these change command line syntax and ABI every minute
  - so libvirt aims to provide
    - stable command interface
    - stable configuraton format
    - stable API
  - configuration format is XML based
  - not all features supported on all hypervisors



Virsh

commandline interface

- virtual machine lifecycle:
    - list, dominfo, dumpxml, define, edit, start, destroy, shutdown
    - save, restore, suspend, resume, migrate
    - domblkstat, domifstat
    - {attach, detach}-{disk, netdev, device}

Virsh

commandline interface

- virtual machine lifecycle:
  - list, dominfo, dumpxml, define, edit, start, destroy, shutdown
  - save, restore, suspend, resume, migrate
  - domblkstat, domifstat
  - {attach, detach}-{disk, netdev, device}
- manage disk images:
  - pool-{list, info, dumpxml, define, edit, start, destroy, refresh}
  - vol-{list, info, create, delete}

Virsh

commandline interface

- virtual machine lifecycle:
  - list, dominfo, dumpxml, define, edit, start, destroy, shutdown
  - save, restore, suspend, resume, migrate
  - domblkstat, domifstat
  - {attach, detach}-{disk, netdev, device}
- manage disk images:
  - pool-{list, info, dumpxml, define, edit, start, destroy, refresh}
  - vol-{list, info, create, delete}
- networks:
  - net-{list, info, dumpxml, define, edit, start, destroy}

Virsh

commandline interface

- virtual machine lifecycle:
  - list, dominfo, dumpxml, define, edit, start, destroy, shutdown
  - save, restore, suspend, resume, migrate
  - domblkstat, domifstat
  - {attach, detach}-{disk, netdev, device}
- manage disk images:
  - pool-{list, info, dumpxml, define, edit, start, destroy, refresh}
  - vol-{list, info, create, delete}
- networks:
  - net-{list, info, dumpxml, define, edit, start, destroy}
- host devices:
  - nodedev-{list, dumpxml}

Programming interface

virsh {list, info, dumpxml, define, start, destroy, shutdown}

Programming interface

virsh {list, info, dumpxml, define, start, destroy, shutdown}

```
int virConnectListDomains (virConnectPtr conn,
                            int *ids, int maxids);
int virDomainGetInfo       (virDomainPtr domain,
                            virDomainInfoPtr info);
char* virDomainGetXMLDesc (virDomainPtr domain,
                            int flags);
virDomainPtr virDomainDefineXML
                           (virConnectPtr conn,
                            const char *xml);
int virDomainCreate        (virDomainPtr domain);
int virDomainDestroy       (virDomainPtr domain);
int virDomainShutdown      (virDomainPtr domain);
```

Language bindings

- C
  - libvirt-dev
  - libvirt-glib-dev
- Python
  - python-libvirt
  - python-libvirt-glib
- Perl
  - libsys-virt-perl
- OCaml
  - libvirt-ocaml
- Ruby
  - libvirt-ruby
- Java
  - not yet packaged in Debian

An example

vmapplet.py

```
import libvirt







conn = libvirt.open("qemu:///system")
ids = conn.listDomainsID()
names = ( conn.lookupByID(id).name()
          for id in ids )
```

An example

vmapplet.py

```
import libvirt



...
def domainEventCallback (conn, dom, event,
                         detail, opaque):
    ...

conn = libvirt.open("qemu:///system")
ids = conn.listDomainsID()
names = ( conn.lookupByID(id).name()
          for id in ids )

conn.domainEventRegister(domainEventCallback,
                         None)
```

An example

vmapplet.py

```
import libvirt
import libvirtglib


...
def domainEventCallback (conn, dom, event,
                         detail, opaque):
    ...

conn = libvirt.open("qemu:///system")
ids = conn.listDomainsID()
names = ( conn.lookupByID(id).name()
          for id in ids )
libvirtglib.eventRegister()
conn.domainEventRegister(domainEventCallback,
                         None)
...
gtk.main()
```

Libvirtd

- does all the hard work
- local socket and remote TCP os SSH interface

```
 -------------        ------       --------------
| application |      | QEMU |--+  |   libvirt    |
 -------------        ------   |    - - - - - - -
       |              ------   |  | drivers API  |
      API            | QEMU |--+--|   *QEMU*     |
       |              ------   |  |    Xen       |
 --------------       ------   |  |    LXC       |
|   libvirt    |     | QEMU |--+  |    remote    |
 - - - - - - -        ------       --------------
| drivers API  |                         |
|    QEMU      |                        API
|    Xen       |                         |
|    LXC       |   qemu:///system    ----------
|   *remote*   | ------------------ | libvirtd |
 --------------                      ----------
```

Libvirtd access

- URIs: qemu:///session, lxc:///
- rw access to */var/run/libvirt/libvirt-sock*
  - via *libvirt* group
  - full access
- read only access: list vms, dump XML data:
  rw access to */var/run/libvirt/libvirt-sock-ro*
- or use PolicyKit
- remote:
  - qemu, xen+{ssh, tcp, tls}://host/...
  - supports SASL authentication (Kerberos) and SSL client certificates
  - virt-manager tunnels VNC over ssh

Tools using Libvirt in Debian

- virtinst:
  - virt-install
  - virt-clone
- virt-manager
- virt-viewer
- virt-top
- munin-libvirt-plugins
- libguestfs (packaged, not yet uploaded)

Virt-install - use existing disk image

```
virt-install --connect=qemu:///system    \
             --name=lenny --ram=256      \
             --os-type=linux             \
             --os-variant=debianLenny    \
             --import                    \
             --disk vol=default/lenny.img \
             --network=user,model=virtio \
```

Virt-install - create virtual machines

```
URL=http://ftp.de.debian.org/debian/
IMAGES=/var/scratch/vm/images/

virt-install --connect=qemu:///system --force         \
  --name=lenny --ram=256                              \
  --disk pool=default,size=10,cache=writeback         \
  --disk vol=default/usb-lenny-preseed.img,bus=usb    \
  --network=user,model=virtio                         \
  --location=${URL}/dists/stable/main/installer-i386/ \
  --extra-args="auto=true priority=critical           \
       url=file:///media/./preseed.cfg                \
       preseed/early_command=\"mountmedia disk\""
```

Storage pools

Keep virtual machine images

- types: directory, (network) filesystem, disk, LVM, iSCSI
- each pool has a target (e.g. directory or volume group)
- some have a source (e.g. block device)

Storage pools

- contain volumes
  - dir/fs pool: files
  - LVM pool: logical volumes
  - disk pool: partitions

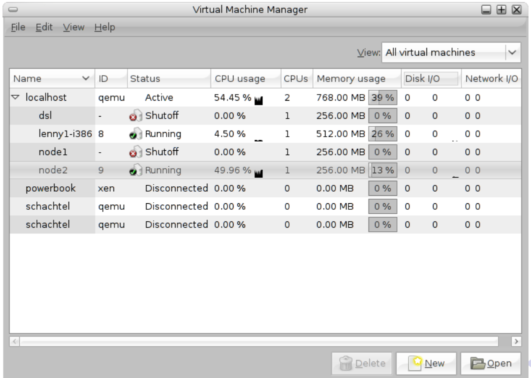
Storage pools

- contain volumes
  - dir/fs pool: files
  - LVM pool: logical volumes
  - disk pool: partitions
- individual volumes (e.g. qcow2 images) can have
  - backing store (potentially read only)

Virt-manager

GUI to manage:

- virtual machines
  - add, remove devices
  - graphical display and serial console
  - display statistics
- pools and volumes
- networks



Name	ID	Status	CPU usage	CPUs	Memory usage		Disk I/O		Network I/O	
localhost	qemu	Active	54.45 %	2	768.00 MB	39 %	0	0	0	0
dsl	-	Shutoff	0.00 %	1	256.00 MB	0 %	0	0	0	0
lenny1-i386	8	Running	4.50 %	1	512.00 MB	26 %	0	0	0	0
node1	-	Shutoff	0.00 %	1	256.00 MB	0 %	0	0	0	0
node2	9	Running	49.96 %	1	256.00 MB	13 %	0	0	0	0
powerbook	xen	Disconnected	0.00 %	0	0.00 MB	0 %	0	0	0	0
schachtel	qemu	Disconnected	0.00 %	0	0.00 MB	0 %	0	0	0	0
schachtel	qemu	Disconnected	0.00 %	0	0.00 MB	0 %	0	0	0	0

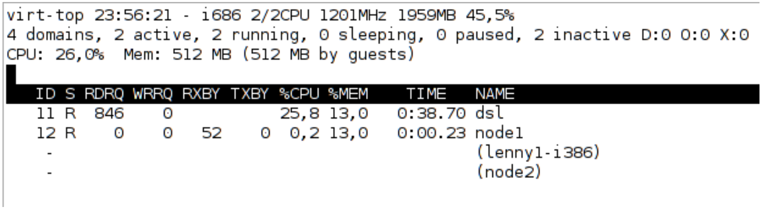
Other virtinst tools

```
$ virt-clone --connect=qemu:///system \
             -o lenny-i386 -n sid-i386
```

```
What would you like to use as the cloned disk (file path) for
'/var/scratch/vm/images/lenny-i386.img'? /var/scratch/vm/images/sid-i386.img
Cloning /var/scratch/vm/i 100% |=========================|  10 GB    02:32
```

Virt-top

Top like tool:



```
virt-top 23:56:21 - i686 2/2CPU 1201MHz 1959MB 45,5%
4 domains, 2 active, 2 running, 0 sleeping, 0 paused, 2 inactive D:0 O:0 X:0
CPU: 26,0%  Mem: 512 MB (512 MB by guests)

   ID S RDRQ WRRQ RXBY TXBY %CPU %MEM    TIME   NAME
   11 R  846    0           25,8 13,0  0:38.70 dsl
   12 R    0    0   52    0  0,2 13,0  0:00.23 node1
    -                                          (lenny1-i386)
    -                                          (node2)
```

Libguestfs - inspect and manipulate vm images

- shell tool: guestfish
- lots of language bindings:
    - libguestfs-perl, python-libguestfs
    - unpackaged: Haskell, Java, Ruby, OCaml

Libguestfs - inspect and manipulate vm images

- shell tool: guestfish
- lots of language bindings:
  - libguestfs-perl, python-libguestfs
  - unpackaged: Haskell, Java, Ruby, OCaml
- recipes: `http://libguestfs.org/recipes.html`
- tools: virt-df, virt-cat, virt-inspector

Example

```
$ guestfish <<EOF
alloc lenny-preseed.img 4M
run
sfdisk /dev/hda 0 0 0 ,
sfdisk-N /dev/hda 1 32 255 63 0,32
mkfs vfat /dev/hda1
mount /dev/hda1 /
upload preseed.cfg /pressed.cfg
umount /
quit
EOF
```

Debugging problems

- /var/log/libvirt, ~/.libvirt/
- virt-manager –nofork, ~/.virt-manager.log
- ~/.virtinst/*.log
- LIBVIRT_DEBUG=1 libvirtd
- LIBGUESTFS_DEBUG=1 guestfish ...

Still not convinced?

Easy to switch to

- domxml-from-native
- `http://wiki.libvirt.org/page/QEMUSwitchToLibvirt`

Missing

- API for disk image snapshots
- fine grained user management
- Debian: SELinux integration

Things to come

- netcf api
- multipath support
- new hypervisors: VMware ESX, OpenNebula
- for QEMU/KVM:
  - netdev hotplug
  - cgroups

Pkg-libvirt group

- Mailing List:
  pkg-libvirt-discuss@lists.alioth.debian.org
- Wiki:
  `http://wiki.debian.org/Teams/DebianLibvirtTeam`