PRESENTED BY RED HAT

LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.

www.theredhatsummit.com

Overview of libvirt and building applications with libvirt

Nandini Chandra
Software Maintenance Engineer
Red Hat
5/5/2011

What is libvirt

- Libvirt library is a set of APIs providing a common layer of abstraction and control for virtual machines,virtual networks and storages
- Components of libvirt: C language API ,a daemon(libvirtd),language bindings for various languages and a command line utility(virsh)
- Cross-vendor project initiated by Red Hat
- Contributors:Community members,IBM,Fujitsu,Novell, Canonical,HP,NEC and other independent developers

What is libvirt

- libvirt is hypervisor independent and provides common functionality that the supported hypervisors implement;supports a number of hypervisors;code is extensible to allow support of other hypervisors;
- KVM
- Xen
- LXC
- OpenVZ
- VMWare ESX and GSX
- UML

What is libvirt

- Multiple hypervisors may run on a single node(eg:LXC and KVM)
- Connection can be established to local node or remote node

Libvirt API

- Provides single-node API(migration is the only multi-node API)
- libvirt 0.7.0 and later is fully thread-safe
- Portable client API for Linux,Solaris,Windows
- Connection has to be established with a hypervisor before using any of the API
- Connection made using one of the three virConnectOpen* calls;a virConnectPtr object is returned by these calls and is used in all subsequent calls made to hypervisor

Objects exposed by libvirt

- virConnectPtr: represents a connection to a hypervisor
- virDomainPtr: represents an active or defined domain
- virNetworkPtr: represents an active or defined network
- virStorageVolPtr: represents a storage volume,
- virStoragePoolPtr: represents a storage pool, i.e. a logical area which can be used to allocate and store storage volumes.
- virInterfacePtr: for configuring host network interfaces (bridging, bonding, VLANs, etc)

Objects exposed by libvirt

- virNWFilterPtr - network filtering
- virNodeDevicePtr - host device passthrough
- virSecretPtr - password and encryption management

Sample Code to create persistent guest

```
/* Compile with gcc -g -Wall sample.c -o sample -lvirt */
#include <libvirt/libvirt.h>

int main(int argc, char *argv[])
{
  virConnectPtr conn;
  virDomainPtr dom;
  const char *xmlconfig = "<domain...XML description...</domain>";

  conn = connectOpenAuth("qemu:///system");
  if (conn == NULL) {
     fprintf(stderr, "Failed to open connection to qemu:///system");
     return;
  }

  dom = virDomainDefineXML(conn, xmlconfig);
  if (!dom) {
      fprintf(stderr, "Failed to define persistent guest configuration");
      return;
  }

  if (virDomainCreate(dom) < 0) {
      VirDomainFree(dom);
      fprintf(stderr, "Failed to boot guest");
  }
}
```


Sample code to create a transient guest

```
virConnectPtr conn;
virDomainPtr dom;
const char *xmlconfig = "<domain...XML description...</domain>";

conn = connectOpenAuth("qemu:///system");
if (conn == NULL) {
  fprintf(stderr, "Failed to open connection to qemu:///system");
  return;
}

dom = virConnectCreateXML(conn, xmlconfig,0);
if (!dom) {
  fprintf(stderr, "Domain creation failed");
  return;
}

fprintf(stderr, “Guest %s has booted”,virDomainName(dom));
virDomainFree(dom);
return;
```


Registering for asynchronous events

- VirEventRegisterImpl : libvirt event loop API allows an application to register for asynchronous events and properly handle them.
- Async events : Booting a guest,restarting a guest,shutdown of a guest,suspension of a guest

Language Bindings

- Libvirt supports C,C++,Python directly.
- Provides bindings for the following languages:C#,Java,OCaml,Perl,PHP,Ruby

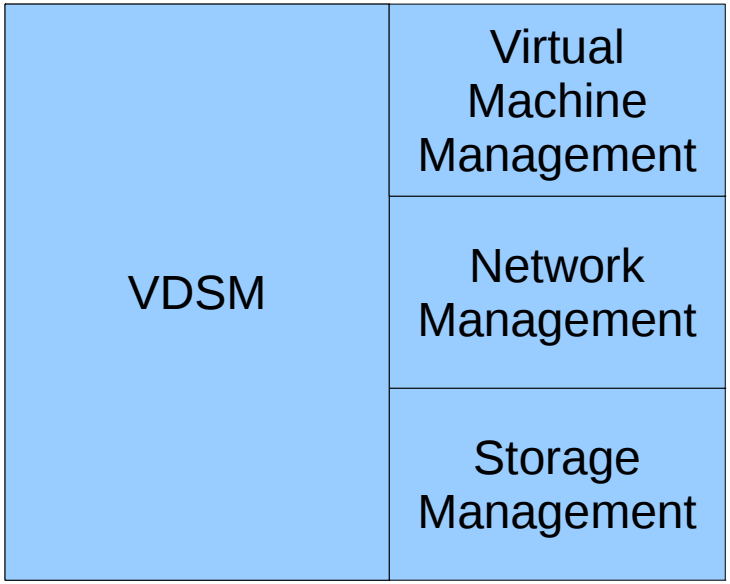
Applications built using libvirt

Command line tools,Web applications,Desktop applications,libraries,configurationManagement, Infrastucture as a Service(IaaS),LiveCD/Appliances; http://libvirt.org/apps.html lists numerous applications built using libvirt

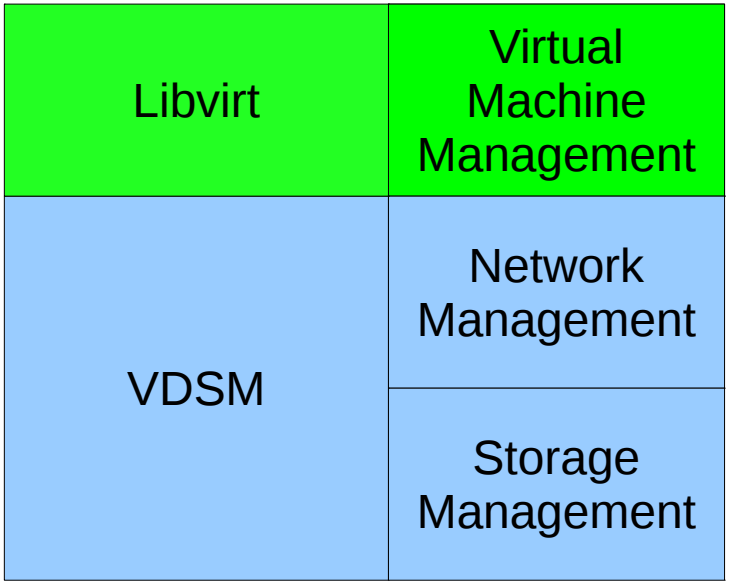
- virsh
- Ovirt
- Virt-manager
- Libguestfs

Applications built using libvirt : RHEV-M

- RHEV-M v 2.x



RHEV-M v 3.0



RHEV-M : Benefits from using libvirt

- RHEV-M has no integrated QEMU driver,hence VDSM is responsible for communicating with QEMU monitor by sending QEMU monitor commands
- QEMU driver is built into Libvirt ; when libvirt is integrated into RHEV-M,libvirt's QEMU driver will communicate with the QEMU monitor
- In RHEV-M,configuration of VMs stored in database,with libvirt configuration stored in XML files;Fetching of configuration information from database generates a lot of traffic

Migration of guests across hosts

- Migration useful for:Load balancing,Hardware Failover,Energy Saving
- Migration can be performed live or offline
- During migration,guest's memory image is sent from source to destination host
- Disk image resides on shared storage
    - Shared Network can use these protocols:NFS,GFS2,iSCSI,Fibre Channel

Implementation of migration

On Source host:

virDomainMigrate

1.conn->driver->domainMigratePrepare2()

Notify Destination host to prepare for migration

2.conn->driver->domainMigratePerform()

```
        if (qemudMonitorCommand (vm, cmd, &info) < 0)
{           qemudReportError (dom->conn, dom,
NULL,VIR_ERR_OPERATION_FAILED,"%s", _("migrate operation
failed"));                  goto cleanup;
                    }
```

3. dconn->driver->domainMigrateFinish2 (remoteDomainMigrateFinish2)

Pass status code to the dst host

Implementation of Migration

On destination host:

1. remoteDispatchDomainMigratePrepare2 (corresponding to 1 on source) virDomainMigratePrepare2
dconn->driver->domainMigratePrepare2 (qemudDomainMigratePrepare2) Destination host notified about the migration,parse the configuration XML of VM,start a qemu-kvm process and listen to a TCP port to receive the memory of VM

2. remoteDispatchDomainMigrateFinish2 (corresponding to 3 on source) virDomainMigrateFinish2
dconn->driver->domainMigrateFinish2 (qemudDomainMigrateFinish2) Receive status code from src.

Nice features to be included in libvirt soon

- Live snapshotting-libvirt currently supports snapshotting of qcow2 images;goal is to expand it to support arbitrary storage types
- KVM and libvirt communities work closely to support KVM ecosystem,to ensure features are implemented across the stack and to bridge gaps between qemu-kvm and libvirt

Resources

- http://libvirt.org/
- IRC:#virt on irc.oftc.net
- Mailing lists:libvirt-users@redhat.com(for users),
  libvir-list@redhat.com(for development)
  http://www.redhat.com/mailman/listinfo/libvir-list
- Application Development Guide
  http://libvirt.org/devguide.html

Questions ?

LIKE US ON FACEBOOK

www.facebook.com/redhatinc

FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

#redhat

READ THE BLOG

summitblog.redhat.com

GIVE US FEEDBACK

www.redhat.com/summit/survey