

Libvirt 安装与使用

——工程实践与科技创新 作业 3

管仁阳-519021911058

实验目的

1. 学习虚拟机管理工具 Libvirt 安装流程
2. 学习使用 Libvirt 管理虚拟机
3. 学习 Libvirt C 语言脚本

实验环境

1. 主机系统: Ubuntu 20.04
2. 虚拟机系统: Ubuntu 20.04
3. qemu 版本: 5.2.0
4. Libvirt 版本: 6.0.0

```
guanrenyang@ubuntu:~/Desktop$ libvirtd
2021-11-24 09:34:58.442+0000: 2833: info : libvirt version: 6.0.0, package: 0ubuntu8.14 (Matthew Ruffell <matthew.ruffell@canonical.com> Tue, 14 Sep 2021 14:00:49 +1200)
2021-11-24 09:34:58.442+0000: 2833: info : hostname: ubuntu
```

相关知识

一、Libvirt 的作用

云平台的实现包括三个部分: (1) 虚拟化技术实现, (2) 虚拟机管理 (3) 集群资源管理, 而使用各种虚拟化技术实现的虚拟机都提供了各自的管理工具。然而, 这种不同虚拟机使用不同工具管理的方式造成了几个问题:

1. 如果采用了混合虚拟化技术, 在上层就需要调用不同的管理工具。
2. 虚拟化技术的发展使得虚拟机的管理工具快速变化。

Libvirt 的目的就是充当抽象管理层, 给上层提供了一套 API, 使之可以使用统一的方式管理使用不同虚拟化方式的虚拟机。

二、Libvirt 管理抽象

1. 连接 (connection): 使用 libvirt 管理虚拟机, 首先要做的就是将 libvirt 与虚拟机监控器 (hypervisor) (在此任务中为 qemu) 建立连接。建立连接以后 libvirt 就可以查找到 hypervisor

下的各个虚拟机。

2. 域 (domain): libvirt 中域就是 hypervisor 下的独立的虚拟机 (virtual machine)。当建立连接以后, 就可以通过域的 ID、名字 (name)、或 UUID 来唯一索引到虚拟机。

三、Libvirt C 数据结构和 API

(一) 数据结构

1. virConnectPtr——表示“连接 (connection)”的类
2. virDomainPtr——表示“域 (domain)”的类
3. virDomainInfo——“域的相关信息”, 包含 5 个数据成员: (1) state: 虚拟机状态 (2) maxMem 最大可用内存 (3) memory: 已用内存 (4) nrVirtCpu: vCPU 数量 (5) cpuTime: cpu 使用时间

(二) 接口

1. virConnectOpenReadOnly(NULL): 以只读方式获取“连接”, 传输 NULL 连接默认 Hypervisor
2. virDomainLookupByID(conn, id): 根据连接 conn 与域 ID id 来查找对应域
3. virDomainGetInfo(dom, &info): 将域 dom 的信息存入 info 内
4. virDomainGetName(dom): 获取域的名字

实验内容

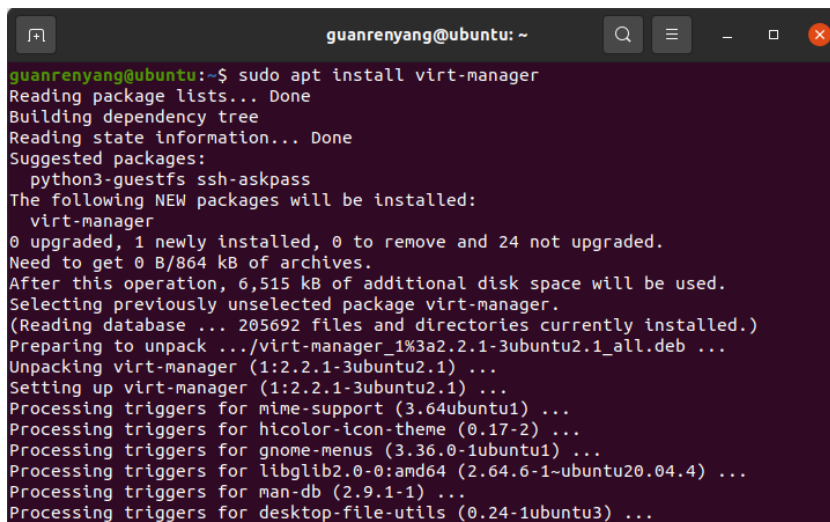
任务一：安装 Libvirt

初次尝试时在官网下载 Libvirt 源码进行配置(configure)编译(compile), 在解决了依赖库不完整的问题以后, 又遇到了 Libvirt 版本与 qemu 版本不一致的问题。

经过查阅资料后发现, Libvirt 包含在 apt 的 virt-manager 包中, 故使用如下指令安装 Libvirt:

```
1 sudo apt install virt-manager
```

安装完成后截图如下:



```
guanrenyang@ubuntu: ~  
guanrenyang@ubuntu:~$ sudo apt install virt-manager  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Suggested packages:  
  python3-guestfs ssh-askpass  
The following NEW packages will be installed:  
  virt-manager  
0 upgraded, 1 newly installed, 0 to remove and 24 not upgraded.  
Need to get 0 B/864 kB of archives.  
After this operation, 6,515 kB of additional disk space will be used.  
Selecting previously unselected package virt-manager.  
(Reading database ... 205692 files and directories currently installed.)  
Preparing to unpack .../virt-manager_1%3a2.2.1-3ubuntu2.1_all.deb ...  
Unpacking virt-manager (1:2.2.1-3ubuntu2.1) ...  
Setting up virt-manager (1:2.2.1-3ubuntu2.1) ...  
Processing triggers for mime-support (3.6ubuntu1) ...  
Processing triggers for hicolor-icon-theme (0.17-2) ...  
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...  
Processing triggers for libglib2.0-0:amd64 (2.64.6-1-ubuntu20.04.4) ...  
Processing triggers for man-db (2.9.1-1) ...  
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
```

使用如下指令查看 Libvirt 版本：

```
1 | libvirt
```

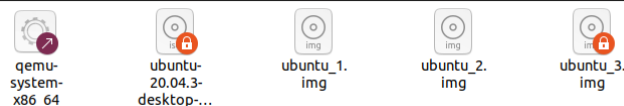
```
guanrenyang@ubuntu:~$ libvirt
2021-11-24 09:42:45.908+0000: 4345: info : libvirt version: 6.0.0, package: 0ubuntu8.14 (Matthew Ruffell <matthew.ruffell@canonical.com> Tue, 14 Sep 2021 14:00:49 +1200)
2021-11-24 09:42:45.908+0000: 4345: info : hostname: ubuntu
```

任务二：编写 C 脚本获取虚拟机 ID, 名称, 最大内存和 vCPU 数量

在编写 C 脚本以前，我们需要使用 Libvirt 创建并启动一个虚拟机。

首先，使用 qemu 创建虚拟机镜像，命名为 ubuntu_3.img

```
1 | sudo apt install virt-manager
```



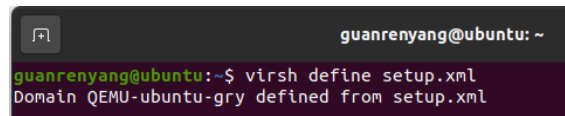
然后，从官方网站 (<https://libvirt.org/drvqemu.html#example-domain-xml-config>) 下载创建虚拟机的 xml 文件。

所有与虚拟机相关的配置都在 setup.xml 文件中完成，在此任务中只需要更改的 1.虚拟机名称 (line2) 2.系统镜像 iso 文件地址 (line14) 3.虚拟机 img 地址 (line19)。其他配置均保持默认配置。

```
Open  setup.xml  Save  [Menu]  [Window]  [Close]
1 <domain type='qemu'>
2   <name>QEMU-ubuntu-gry</name>
3   <uuid>c7a5fdbd-cdaf-9455-926a-d65c16db1809</uuid>
4   <memory>219200</memory>
5   <currentMemory>219200</currentMemory>
6   <vcpu>2</vcpu>
7   <os>
8     <type arch='i686' machine='pc'>hvm</type>
9     <boot dev='cdrom' />
10  </os>
11  <devices>
12    <emulator>/usr/bin/qemu-system-x86_64</emulator>
13    <disk type='file' device='cdrom'>
14      <source file='/home/guanrenyang/qemu-6.1.0/build/x86_64-softmmu/ubuntu-20.04.3-desktop-
amd64.iso' />
15      <target dev='hdc' />
16      <readonly />
17    </disk>
18    <disk type='file' device='disk'>
19      <source file='/home/guanrenyang/qemu-6.1.0/build/x86_64-softmmu/ubuntu_3.img' />
20      <target dev='hda' />
21    </disk>
22    <interface type='network'>
23      <source network='default' />
24    </interface>
25    <graphics type='vnc' port='-1' />
26  </devices>
27 </domain>
```

完成配置 setup.xml 以后，使用如下指令创建虚拟机

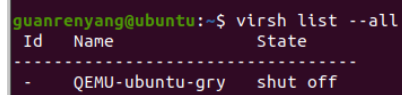
```
1 | virsh define setup.xml
```



```
guanrenyang@ubuntu: ~  
guanrenyang@ubuntu:~$ virsh define setup.xml  
Domain QEMU-ubuntu-gry defined from setup.xml
```

接着可以使用如下指令查看创建的虚拟机:

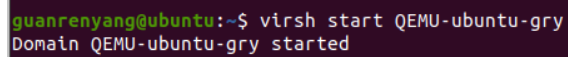
```
1 | virsh list --all
```



```
guanrenyang@ubuntu:~$ virsh list --all  
Id      Name                               State  
-----  
-       QEMU-ubuntu-gry                   shut off
```

此时创建好的虚拟机处在关机状态，使用如下指令打开虚拟机

```
1 | virsh start QEMU-ubuntu-gry
```



```
guanrenyang@ubuntu:~$ virsh start QEMU-ubuntu-gry  
Domain QEMU-ubuntu-gry started
```

完成创建并启动虚拟机后，编写 C 脚本获取虚拟机的相关信息。

C 语言脚本如下所示

```
Terminal
#include<stdio.h>
#include<libvirt/libvirt.h>
#include<cstring>
int main(int argc, char **argv)
{
    int id = argv[1][0] - '0';

    virConnectPtr conn = NULL;
    virDomainPtr dom = NULL;
    virDomainInfo info;
    char name[20];

    conn = virConnectOpenReadOnly(NULL);
    if (conn == NULL) {
        fprintf(stderr, "Failed to connect to hypervisor\n");
        return 1;
    }

    dom = virDomainLookupByID(conn, id);
    if (dom == NULL) {
        fprintf(stderr, "Failed to find Domain %d\n", id);
        virConnectClose(conn);
        return 1;
    }

    if (virDomainGetInfo(dom, &info) < 0) {
        fprintf(stderr, "Failed to get information for Domain %d\n", id);
        virDomainFree(dom);
        virConnectClose(conn);
        return 1;
    }

    strcpy(name, virDomainGetName(dom));

    printf("ID: %d\n", id);
    printf("Name: %s\n", name);
    printf("maxMem: %ld KB\n", info.maxMem);
    printf("vCPUs: %d\n", info.nrvirtCpu);

    if (dom != NULL){
        virDomainFree(dom);
    }
    if (conn != NULL){
        virConnectClose(conn);
    }
    return 0;
}
```

编译运行程序，结果如下所示：

```
guanrenyang@ubuntu:~$ ./libvirt_c_api 1
ID: 1
Name: QEMU-ubuntu-gry
maxMem: 220160 KB
vCPUs: 2
```