

上海交通大學

SHANGHAI JIAOTONG UNIVERSITY



工程实践与科技创新 3-D 作业 3

Python 脚本获取虚拟机信息

姓名：薛春宇

学号：518021910698

完成时间：2020/10/9

Python 脚本调用 Libvirt 获取虚拟机信息

薛春宇 518021910698

1. 实验目的

安装 Libvirt，并利用 Python 脚本调用 Libvirt 的 API 来获取虚拟机的 ID、名称、虚拟机启动状态、最大内存和虚拟 CPU 个数。

2. 实验内容

2.1 Libvirt 基本原理

Libvirt 是用于管理虚拟化平台的开源的 API，后台程序和管理工具。它可以用于管理 KVM、Xen、VMware ESX，QEMU 和其他虚拟化技术。这些 API 在云计算的解决方案中广泛使用。

Libvirt 是提供了一个方便的方式来管理虚拟机和其他虚拟化功能的软件的集合，如存储和网络接口管理。这些软件包括一个 API 库，一个守护进程（Libvirtd），和一个命令行实用程序（virsh）。

Libvirt 的首要目标是能够管理多个不同的虚拟化供应商/虚拟机管理程序提供一个单一的方式。例如，命令“virsh 列表等等都可以用于任何支持现有的虚拟机管理程序列表（KVM、Xen、VMware ESX，等）不需要学习管理程序特定的工具。

2.2 Libvirt 的使用

2.2.1 利用 virt-manager 安装虚拟机

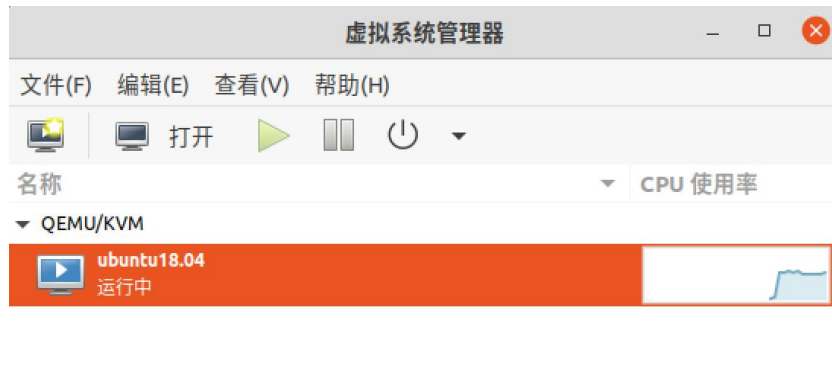
本次虚拟机使用的是 Ubuntu20.04 版本。首先，使用指令安装 qemu 环境：

```
sudo apt-get install qemu
sudo apt-get install qemu-kvm
```

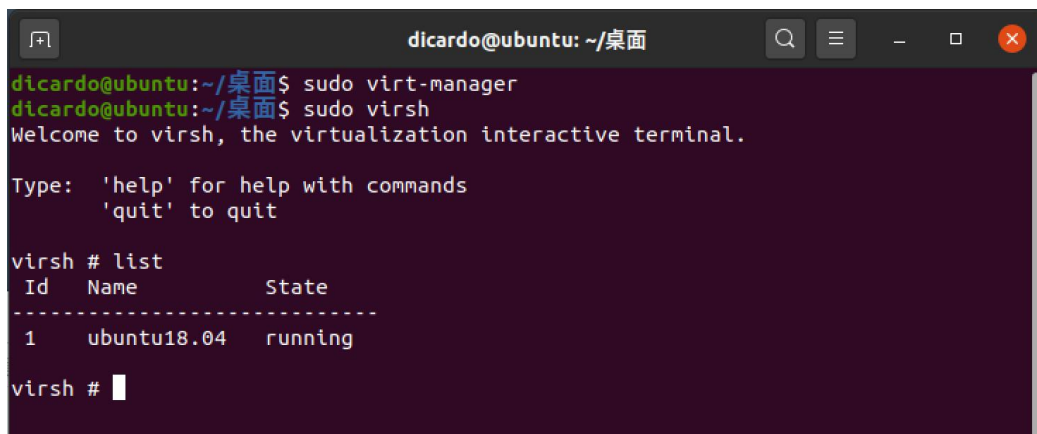
安装相关软件和包：

```
sudo apt-get install virtinst
sudo apt-get install libvirt-dev
sudo apt-get install libvirt-bin
```

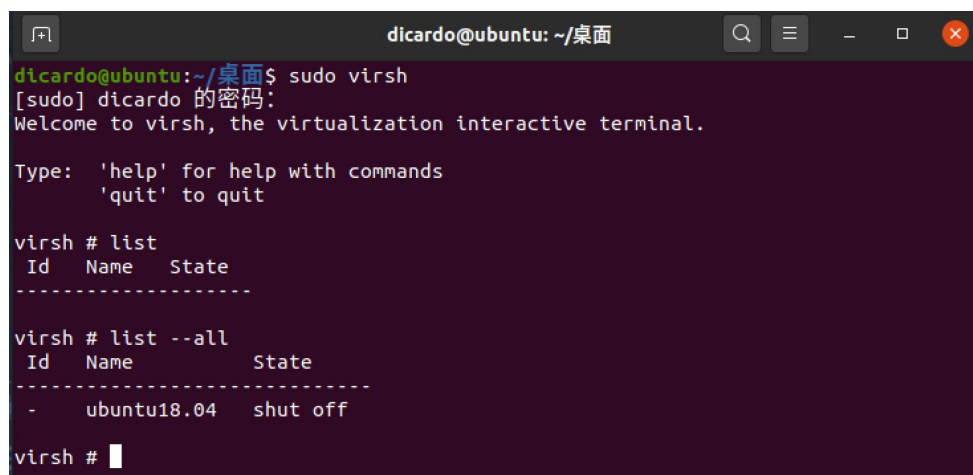
使用 `sudo virt-manager` 指令打开虚拟机管理器, 并新建一个 Ubuntu18.04 虚拟机并运行:



可以使用 `sudo virsh` 指令进入 virsh 环境, 并使用 `list` 指令查看运行过程中的虚拟机:



也可以在 virsh 环境中使用 `list --all` 指令查看包括已关闭虚拟机在内的全部虚拟机:



2.2.2 利用 python 脚本获取虚拟机信息

在虚拟机运行过程中运行如下脚本代码：

```
import libvirt
conn = libvirt.open("qemu:///system")
for id in conn.listDomainsID():
    domain = conn.lookupByID(id)

    print ("The ID of VM: %s" % (domain.UUIDString()))
    print ("The Name of VM: %s" % (domain.name()))
    print ("The Status of VM: %d (1 for running)" % (domain.info()[0]))
    print ("The Max Memory of VM: %d MB" % (domain.info()[1]))
    print ("The Number of Virtual CPUs: %d" % (domain.info()[3]))
conn.close()
```

注意，在执行脚本时报错：

```
dicardo@ubuntu:~/桌面$ python check.py
Traceback (most recent call last):
  File "check.py", line 1, in <module>
    import libvirt
ImportError: No module named libvirt
dicardo@ubuntu:~/桌面$
```

解决方案是安装 python3-libvirt 包：

```
sudo apt-get install python3-libvirt
```

利用 python3 指令运行脚本：

```
python3 check.py
```

注意，运行过程中要保持 virt-manager 中的虚拟机一直处于非关机状态，否则脚本不会有输出。

成功运行脚本后的运行结果如下：

```
dicardo@ubuntu: ~/桌面
dicardo@ubuntu:~/桌面$ python3 check.py
The ID of VM: 0ac6f87a-190d-46f7-b233-cac4d5821a6a
The Name of VM: ubuntu18.04
The Status of VM: 1 (1 for running)
The Max Memory of VM: 2097152 MB
The Number of Virtual CPUs: 2
dicardo@ubuntu:~/桌面$
```

可以看到，该虚拟机的 ID 为 0ac6f87a-190d-46f7-b233-cac4d5821a6a，名称为 ubuntu18.04，当前处于运行状态，最大内存为 2097152MB，有 2 个虚拟 CPU。

3. 实验结论

本次实验中，我们实现了以下内容：

- (1) 安装 Libvirt;
- (2) 编写 python 脚本实现对虚拟机监控信息的抓取及展示;

4. 实验心得

在刚刚开始本次实验时，我根据网上搜得的资料，对 libvirt 源码进行了编译和安装，在过程中也遇到了非常多的问题。在我的不断努力之下，这些问题一一被解决：

Test

安装过程

libnl-dev依赖包补充

No package 'libnl-route-3.0' found [解决方案](#)

configure: error: You must install the libyajl library & headers to compile libvirt [解决方案](#)

./configure: 41: ./configure: cmake: not found

需要安装cmake:

```
sudo apt-get install cmake
sudo apt-get install cmake-qt-gui
```

之后分别执行：

```
./configure
make
sudo make install
```

(makefile发生错误后) 更改为 libvirt-4.4.0 版本.

./configure 遇到 configure: error: "xsltproc is required to build libvirt" 错误。解决方案 `sudo apt-get install xsltproc`

完成 ./configure 后开始 make ,完成。

之后开始 `sudo make install` ,完成。

reboot 重启。

下载 Ubuntu18.04 镜像，其间顺便安装一下 `sudo apt-get install virt-manager` 。一直报错，决定从一开始就先安装好 virt-manager .

./configure 遇到 configure: error: libnl-devel >= 1.1 is required for macvtap support 错误。解决方案参见[libnl-dev依赖包补充](#)

No package 'libnl-route-3.0' found 解决方案见上。

configure: error: libxml2 >= 2.6.0 is required for libvirt 错误。 [解决方案](#) (一共两个指令)。

configure: error: You must install the pciaccess module to build with udev 错误。 [解决方案](#)

configure: error: You must install device-mapper-devel/libdevmapper to compile libvirt with mpath storage driver

解决方法: `sudo apt-get install libdevmapper-dev` [解决方案](#)

`./configure` 完成, 开始 `make`

(放弃从源码编译安装)

```
sudo apt-get install virt-manager
sudo apt-get install virtinst
sudo apt-get install libvirt-bin
sudo apt-get install libvirt-dev
```

由于不确定 ubuntu20.04 到底适配哪个版本的 libvirt, 我分别尝试了 1.1.2、1.2.5 和在经过约半天时间的尝试之后, 我发现这样的实现方式过于繁杂且不实用。后来发现, 本次作业的要求仅仅是安装 libvirt, 而不是从源码进行编译和安装, 因此决定直接利用 apt 软件库进行安装。

总的来说, 我们很好地实现了本次实验的要求。虽然过程中遇到了很多困难, 也走了些弯路, 但我认为这些尝试都是值得的, 这让我学会了不少东西!