

课程大作业二报告

519021910029-刘永翔

一、下载 QEMU5.2.0，并且编译

由于最新版本是 6.1.0，因此想要下载安装 5.2.0 版本的 QEMU，就需要从官网上下载 tar 压缩文件，并且解压、编译。

查看 QEMU 的 introduction 发现，这是一款针对于 linux 系统所设计的软件，在 windows 系统下还不成熟，因此我选择在大作业一中所安装的 ubuntu 虚拟机中安装 QEMU。

```
wget https://download.qemu.org/qemu-6.1.0.tar.xz
tar xvf qemu-6.1.0.tar.xz
cd qemu-6.1.0
./configure
make
```

过程如上图所示，只要将 6.1.0 修改成 5.2.0 即可。需要注意的是，我在虚拟机中使用 wget 下载压缩包时，发现下载速度很慢。所以我先在 windows 系统下将该压缩包下载下来，再利用 WinSCP 软件通过连接虚拟机的 ip 来进行文件传输，将压缩包传输到虚拟机中，再进行接下来的操作。

解压没有遇到问题，在运行 ./configure 时，需要额外安装一些包，来辅助接下来的编译工作。

它们分别对应的指令是：

```
sudo apt-get install ninja-build
sudo apt-get install gcc
sudo apt-get install build-essential zlib1g-dev pkg-config libglib2.0-dev
binutils-dev libboost-all-dev autoconf libtool libssl-dev
libpixman-1-dev libpython-dev python-pip python-capstone virtualenv
sudo apt-get install libpixman-1-dev
```

依次运行完成后，运行 make 便可开始编译，等待一个小时左右编译完成！

二、创建 Tap mode 两个虚拟机，分别使用 e1000 和 virtio-net 两种模型

创建虚拟机之前，要先下载相对应的虚拟机的镜盘映像，考虑到虚拟机下载速度较慢，和步骤一中一样，我先将 ubuntu18.04 的映像通过 WinSCP 传输到了虚拟机上。接下来就可以在虚拟机上，通过 QEMU 再创建两个虚拟机了。

通过查阅资料，我了解到，利用 QEMU 创建的虚拟机一般分为两个模式，一种是 User Mode，一种是 Tap Mode。

想要创建一个 Tap Mode 的虚拟机，我们就先要创建一个 tap，除此之外我们还需要给这个虚拟机一个特定的网卡。

在虚拟机中创建 Tap Mode 虚拟机时，我们必须要先搭建一个网桥，通过这个网桥，将虚拟机的网卡和 tap 通过网桥连接起来。通过 ifconfig 命令可以查到，我的虚拟机上的网卡的名称为 eno33，因此在创建这两个 tap mode 虚拟机前需要执行一下指令：

```
sudo apt-get install bridge-utils //用于获取网桥管理工具 brctl
sudo apt-get install uml-utilities//用于获取创建 tap 的工具 tuncctl
```

```

sudo brctl addbr br0
sudo ip addr flush dev eno33
sudo brctl addif br0 eno33
sudo tuncctl -t tap1 -u 'samliu'//samliu 是我虚拟机的用户名
sudo tuncctl -t tap2 -u 'samliu'//分别为两个虚拟机创建两个 tap
sudo addif br0 tap1//将 br0 和 tap1 相关联
sudo addif br0 tap2
sudo ifconfig eno33 up
sudo ifconfig tap1 up
sudo ifconfig tap2 up
sudo ifconfig br0 up
sudo dhclient -v br0//给 br0 分配 ip

```

在这个地方需要注意的是，dhclient 这条命令只有在虚拟机中才能够顺利使用，我原本希望在自己另外安装的一个 ubuntu 系统下，使用这条命令来获得 ip 地址并安装两个虚拟机，发现这条指令会卡死而无法获得动态分配的 ip！

由此网桥和 tap 创建完毕，我们可以开始创建虚拟机了。

安装 ubuntu 虚拟机至少要给他分配 10G 的硬盘内存，为了方便管理，我希望这个虚拟机会安装在一个 img 文件中。因此使用一下指令创建两个 10G 的 img 供其安装：

```

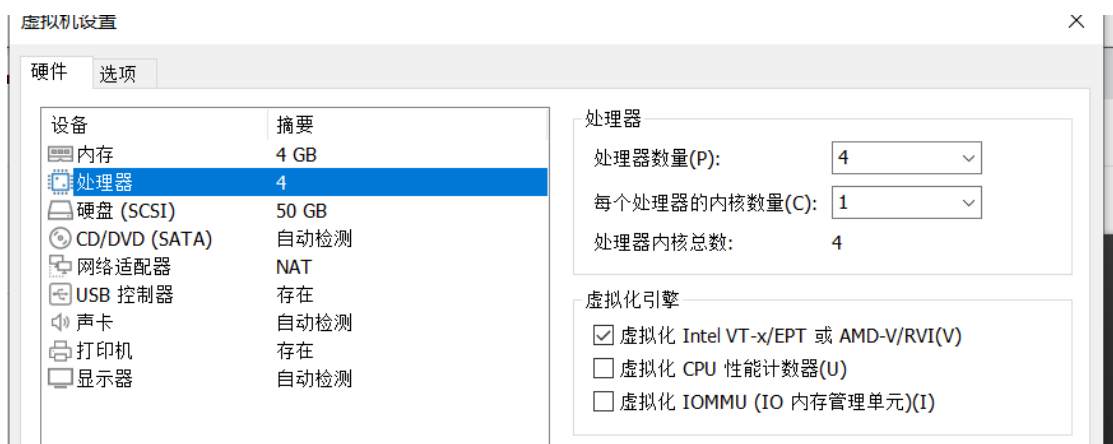
sudo qemu-img create -f qcow2 VM1.img 10G
sudo qemu-system-x86_64 VM1.img
sudo qemu-img create -f qcow2 VM2.img 10G
sudo qemu-system-x86_64 VM2.img

```

此时需要注意一点，在 VMware 上的 host 虚拟机硬盘内存要给足，否则不足以支持他安装两个 ubuntu 虚拟机，当内存不足时会出现安装失败或无法再次打开 host 虚拟机的情况。

在正式开始安装前，我们需要查看虚拟机是否有硬件虚拟化支持，输入：lsmod | grep kvm 后发现并没有输出，因此尚未打开虚拟化支持。

打开虚拟化支持，我们只需要在 VMware 中勾选一项即可，如下图所示



在此之后，我希望在 VM1 中安装 e1000 网卡模型，VM2 中安装 virtio-net 网卡模型，分别使用以下命令，创建后安装 ubuntu 系统即可实现：

```

sudo qemu-system-x86_64 -m 2048 -enable-kvm VM1.img -cdrom ../ubuntu-18.04.5-desktop-amd64.iso -boot d -netdev tap,id=mynet1,ifname=tap1,script=no,downscript=no -device

```

```
e1000,netdev=mynet1,mac=52:55:00:d1:55:01
```

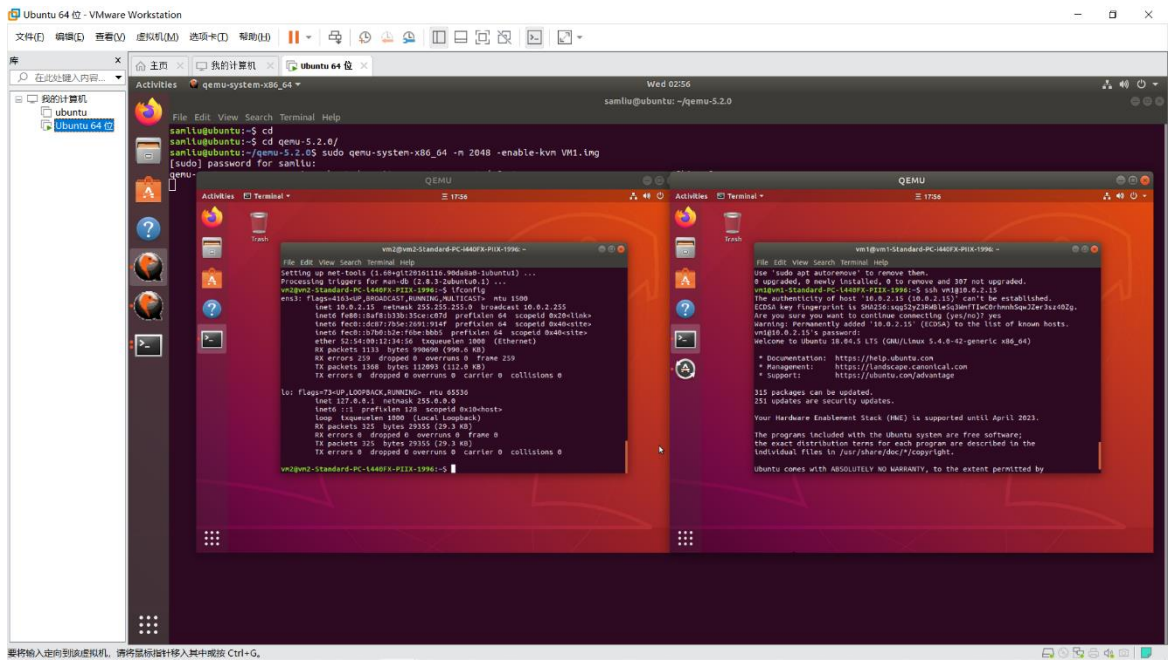
```
sudo qemu-system-x86_64 -m 2048 -enable-kvm VM2.img -cdrom ../ubuntu-18.04.5-desktop-amd64.iso -boot d -netdev tap,id=mynet2,ifname=tap2,script=no,downscript=no -device virtio-net,netdev=mynet2,mac=52:55:00:d1:55:02
```

三、使用 ssh 将两个虚拟机相连接

在使用 ssh 之前，先要在两个虚拟机上安装 ssh-server，运行：

```
sudo apt-get install ssh-server
```

再利用 ifconfig 命令获得 ip 地址即可使用 ssh 进行连接



如上图所示，vm2 的 inet 地址为 10.0.2.15，因此在 vm1 中，输入：

ssh vm2@10.0.2.15，输入密码后，即可进行连接。vm2 连接 vm1 同理。

在运行 ifconfig 时，我发现二者的 inet 地址是一样的，究其原因，应该是因为二者的网卡都和同一个网桥相连接，因此拥有了相同的 inet 地址，在连接 ssh 时只需要将 @ 前面的用户名改一下就好了。

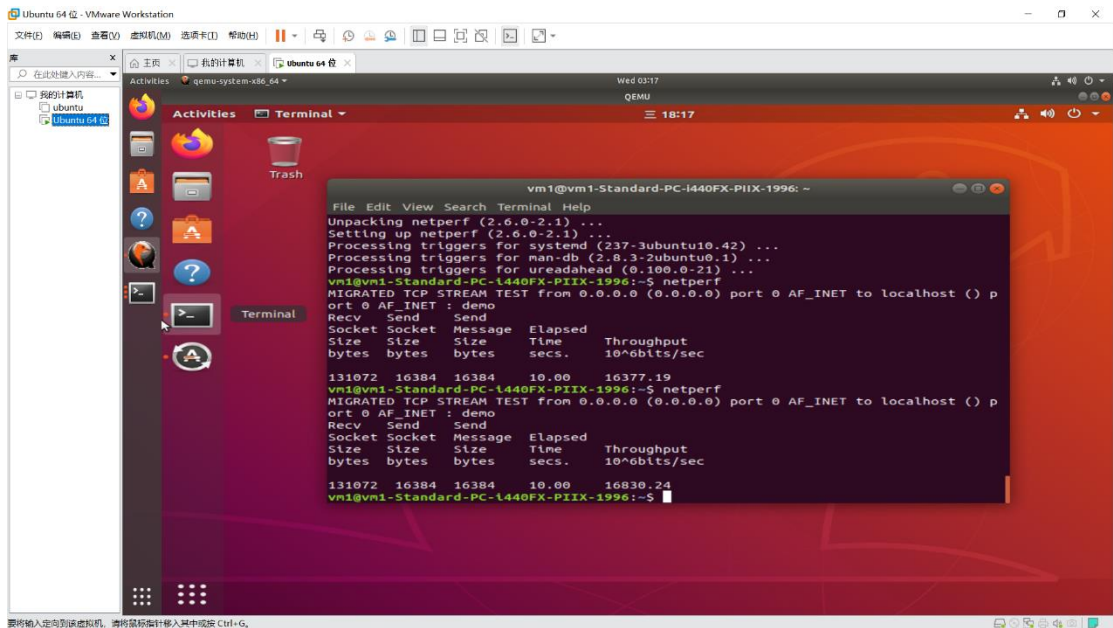
四、测试虚拟机和主机的网络性能

测试网络性能，使用到的工具就是现成的工具包 netperf。先安装 netperf：

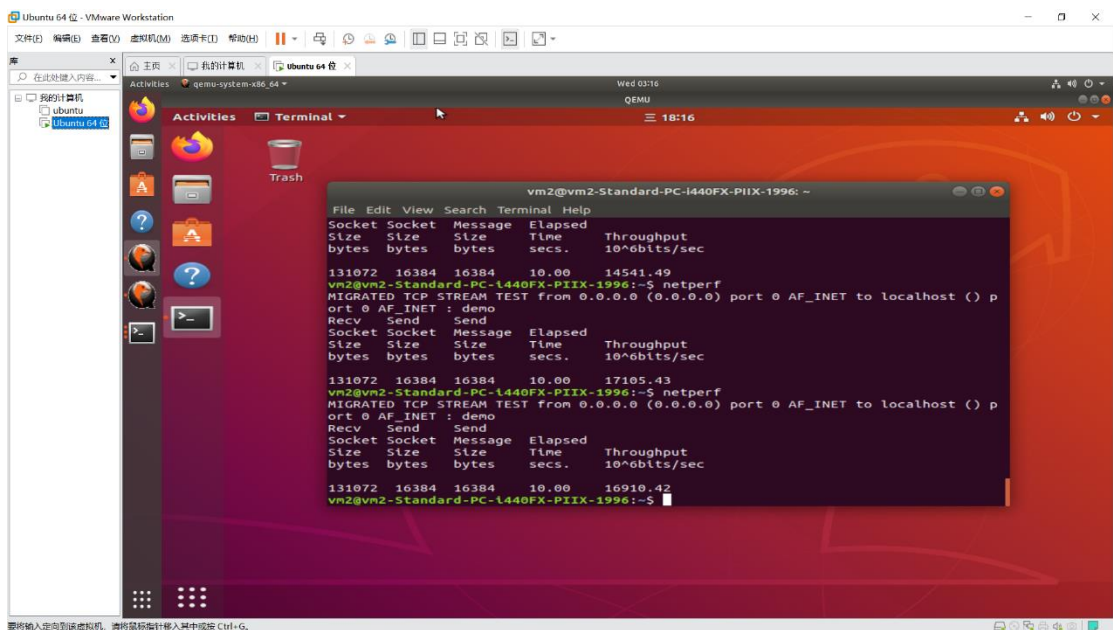
```
sudo apt-get install netperf
```

在三个机器上分别运行 netperf 得到结果如下：

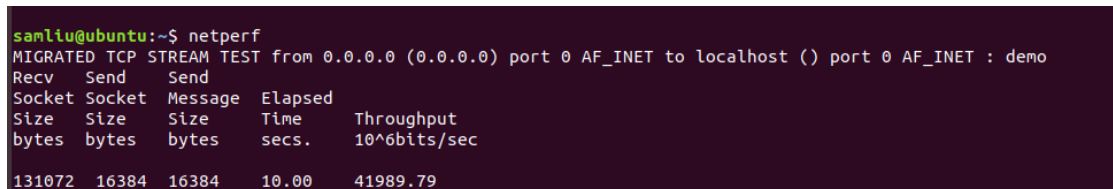
e1000 VM: throughput \approx 2.10GB/s



virtio-net VM: throughput \approx 2.15GB/s, 略高于 e1000



Host: throughput \approx 5.25GB/s



通过对比发现，Host 的网络性能最好，明显高于两个虚拟机。而在两个虚拟机中，virtio-net 的虚拟机网络性能略强于 e1000 的网络性能。究其原因，我认为是两个 VM 使用了两个不同的 tap，像是并联电路分流了 Host 上的 throughput，但是也略有损耗，导致出现这个结果。

五、总结

此次大作业让我们安装了 QEMU，并学习了如何利用 QEMU 创建 tap mode 的虚拟机，如何使用 ssh 连接虚拟机和如何测试设备的网络性能。新的软件、环境意味着新的挑战，在这次大作业中，我除了学习到了上述知识以外，更重要的是增强了自己的自学能力，更加善于阅读指导文件、查询问题、解决问题，收获颇丰！