

Parallel Desktop 的安装和使用

学号： 姓名： 指导教师：

1 实验环境和相关配置

- 1. 宿主操作系统：macOS
- 2. 虚拟机软件：Parallel Desktop 19 Business Edition
- 3. 虚拟机操作系统：Ubuntu 20.04
- 4. 本实验使用 macOS 系统，可以为以后使用 macOS 的同学提供一些参考；鉴于实验参考书中使用的 Centos 操作系统，本实验报告提供一些 Ubuntu 的使用方法

2 实验目的和要求

- 1. 安装 Parallel Desktop，并且完成相关的配置。
- 2. 建立虚拟机集群，首先要求能上网，掌握如何创建、克隆和管理虚拟机。要求建立三个虚拟机，实现相互发送和接收数据包。
- 3. 了解 SSH 的基本使用。要求实现三台机器之间的无密码互访，如，master 可以无密码远程登录 slave1 和 slave2。
- 4. 使用 ssh secure file transfer cliet 实现 windows 主机与 linux 虚拟机文件互传，使用 ssh sucure shell client 实现在 Windows 上远程登录访问服务器。

3 实验内容

3.1 安装 Parallel Desktop

从官网下载 Parallel Desktop 的镜像文件，然后进行安装，基本全为默认。

3.2 建立和克隆虚拟机

Parallel Desktop 的虚拟机安装过程较为便捷，如果想要快速安装可以直接选取图1中提供的相关系统，直接进行安装，安装默认的是最新版本，如果想要使用之前版本，可以在官网下载镜像然后通过镜像进行安装。

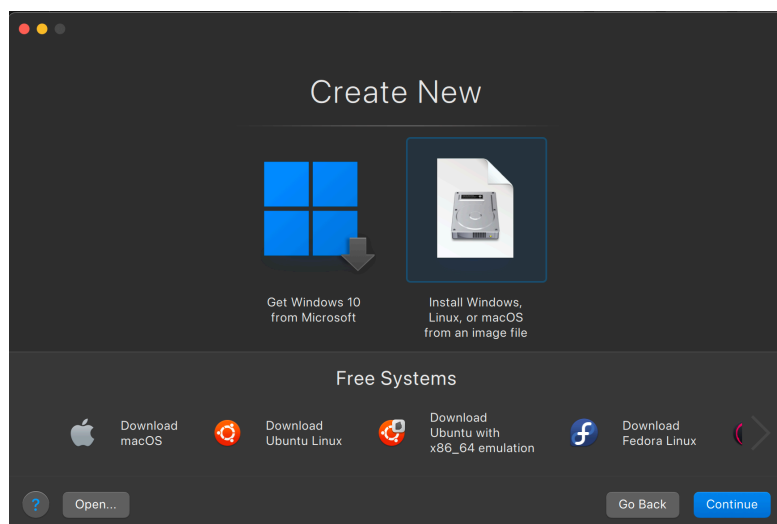


图 1: Parallel Desktop 的虚拟机安装界面

安装时还可以调整分配给虚拟机的硬件资源，如内存、CPU 核心数、磁盘空间等，如图2所示，如果有个人的需要可以进行调整否则直接全部默认即可。

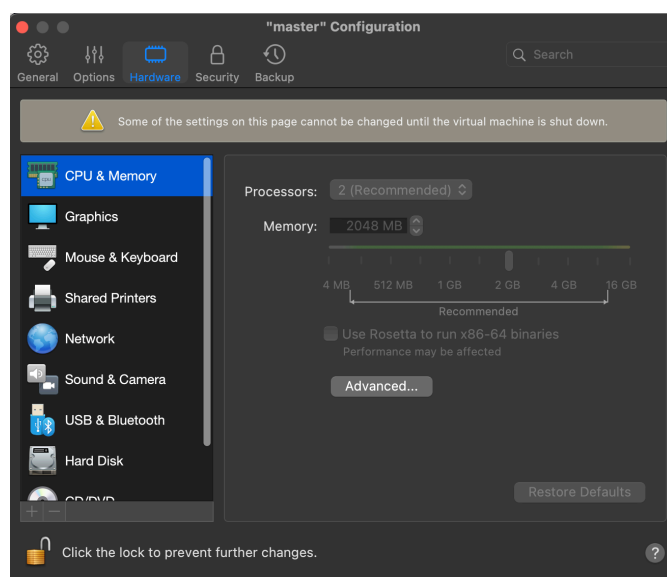


图 2: Parallel Desktop 的虚拟机配置界面

安装完成后，可以进行虚拟机的克隆，如图3所示，直接复制两个相同的虚拟机然后双击添加到 Parallel Desktop 中即可。

表 1: 静态 IP 分配表

虚拟机名称	IP 地址	子网掩码	网关
master	192.168.1.10	255.255.255.0	192.168.1.1
slave1	192.168.1.11	255.255.255.0	192.168.1.1
slave2	192.168.1.12	255.255.255.0	192.168.1.1

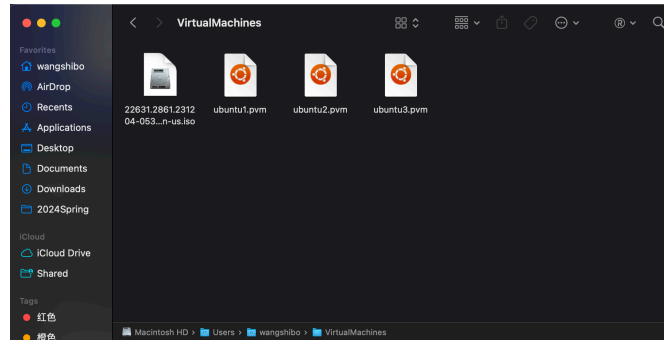


图 3: 克隆虚拟机

4 配置虚拟机网络

虚拟机集群的组网方式使用了 Parallel Desktop 提供的共享网络模式，在 VMware 中也被称为 NAT 模式。在这种模式下，宿主电脑和虚拟机集群在同一个局域网中（内网），介入到这个局域网的网卡使用私有 IP 地址进行相互的通信，并且宿主电脑相当于路由器，起到网关的作用，虚拟机通过宿主电脑访问互联网。

如果想要创建虚拟机集群，那么就要使用静态 IP 分配，不可以使用 DHCP（动态 IP 分配），因为要固定虚拟机之间的通信地址，如果每一次重启虚拟机集群或者宿主机都会动态分配 IP 地址那么就需要重新配置虚拟机集群的网络。拟使用下面的静态 IP 分配方式，宿主电脑提供一个 192.168.1.0/24 的网段，最多可供 253 台虚拟机使用（除去全 0 的网段，为 1 的网关，全为 1 的广播地址）。

配置虚拟机的网络需要配置如下两个方面（三台机器的配置几乎相同，这里只介绍 master 的配置）：

- 1. **配置虚拟机的静态 IP 地址：**Ubuntu 的网络配置文件存放位置不同于 Centos，存放在 /etc/netplan/目录下，文件名为 00-installer-config.yaml，首先使用 ifconfig 指令查看链接局域网的网卡名称，然后使用 vim 编辑器打开 00-installer-config.yaml 文件，修改相关内容，在进行之前需要先使用 cp 命令对文件进行备份，修改完成后使用 netplan apply 使得修改生效并且使用 ifconfig 查看是否修改成功。下面是修改的相关代码和文件内容。

```

1 parallels@master:~$ sudo su
2 [sudo] password for parallels:
3 root@master:/home/parallels# ifconfig -a
4 enp0s5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
5     inet 10.200.1.10 netmask 255.255.255.0 broadcast 10.200.1.255
6     inet6 fe80::21c:42ff:febf:5095 prefixlen 64 scopeid 0x20<link>
7     inet6 fdb2:2c26:f4e4:0:c36c:c129:57e0:4392 prefixlen 64 scopeid 0x0<global>

```

```

8      inet6 fdb2:2c26:f4e4:0:21c:42ff:febf:5095 prefixlen 64 scopeid 0x0<global>
9      ether 00:1c:42:bf:50:95 txqueuelen 1000 (Ethernet)
10     RX packets 2344 bytes 772011 (772.0 KB)
11     RX errors 0 dropped 0 overruns 0 frame 0
12     TX packets 915 bytes 94756 (94.7 KB)
13     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
14 root@master:/home/parallels# cd /etc/netplan
15 root@master:/etc/netplan# cp 00-installer-config.yaml 00-installer-config.yaml_backup
16 root@master:/etc/netplan# vim 00-installer-config.yaml
17 root@master:/etc/netplan# netplan apply
18 ** (generate:32379): WARNING **: 14:21:39.983: `gateway4` has been deprecated, use
    default routes instead.
19 See the 'Default routes' section of the documentation for more details.
20 root@master:/etc/netplan# ifconfig -a
21 enp0s5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
22     inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
23     inet6 fe80::21c:42ff:febf:5095 prefixlen 64 scopeid 0x20<link>
24     ether 00:1c:42:bf:50:95 txqueuelen 1000 (Ethernet)
25     RX packets 2361 bytes 774788 (774.7 KB)
26     RX errors 0 dropped 0 overruns 0 frame 0
27     TX packets 975 bytes 101247 (101.2 KB)
28     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
29

```

Listing 1: 配置静态 IP 指令

```

1 network:
2 version: 2
3 renderer: NetworkManager
4 ethernets:
5     enp0s5:
6         dhcp4: no
7         dhcp6: no
8         addresses: [192.168.1.10/24]
9         gateway4: 192.168.1.1
10        nameservers:
11            addresses: [8.8.8.8, 114.114.114.114]
12

```

Listing 2: 修改后的 00-installer-config.yaml 文件内容

可以看到上面的虚拟机的 IP 地址，网关已经修改，对剩下的两台虚拟机做同样的操作即可。

- **2. 配置虚拟机的 hostname 和 IP 地址的映射关系：**对于虚拟机集群之间，可以直接使用 IP 地址进行通信，但是对于程序员，还是给每一个电脑取一个名字比较好记忆和维护，所以需要配置 hostname 和 IP 地址的映射关系，这里使用/etc/hosts 文件进行配置。

```

1 root@master:/etc/netplan# cd /
2 root@master:/# cd /etc
3 root@master:/etc# vim hostname
4 root@master:/etc# hostname
5 master
6 root@master:/etc# vim hosts
7

```

Listing 3: 配置 hostname 和 IP 地址的映射关系指令

```

1 127.0.0.1 localhost
2 127.0.1.1 ubuntu-linux-22-04-desktop
3 192.168.1.10 master
4 192.168.1.11 slave1
5 192.168.1.12 slave2
6 # The following lines are desirable for IPv6 capable hosts
7 ::1      ip6-localhost ip6-loopback
8 fe00::0  ip6-localnet
9 ff00::0  ip6-mcastprefix
10 ff02::1  ip6-allnodes
11 ff02::2  ip6-allrouters
12

```

Listing 4: 修改后的 hosts 文件内容

对剩下的两台虚拟机做同样的操作，然后就可以开始测试了。

5 测试虚拟机集群对互联网和内网虚拟机之间的访问性

首先测试虚拟机集群对互联网的访问性，使用 ping 命令给 baidu.com 发送 ICMP 报文，然后在使用 ping 命令测试对 slave1 和 slave2 的访问性，如图4所示，可见完成了实验目标。

```

root@master:/home/parallels# ping -c 5 baidu.com
PING baidu.com (110.242.68.66) 56(84) bytes of data.
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=1 ttl=128 time=1.24 ms
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=2 ttl=128 time=0.540 ms
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=3 ttl=128 time=1.44 ms
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=4 ttl=128 time=1.59 ms
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=5 ttl=128 time=1.09 ms

--- baidu.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4017ms
rtt min/avg/max/mdev = 0.540/1.179/1.585/0.361 ms
root@master:/home/parallels# ping -c 5 slave1
PING slave1 (192.168.1.11) 56(84) bytes of data.
64 bytes from slave1 (192.168.1.11): icmp_seq=1 ttl=64 time=0.398 ms
64 bytes from slave1 (192.168.1.11): icmp_seq=2 ttl=64 time=0.632 ms
64 bytes from slave1 (192.168.1.11): icmp_seq=3 ttl=64 time=0.566 ms
64 bytes from slave1 (192.168.1.11): icmp_seq=4 ttl=64 time=0.539 ms
64 bytes from slave1 (192.168.1.11): icmp_seq=5 ttl=64 time=0.616 ms

--- slave1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4105ms
rtt min/avg/max/mdev = 0.398/0.550/0.632/0.083 ms
root@master:/home/parallels# ping -c 5 slave2
PING slave2 (192.168.1.12) 56(84) bytes of data.
64 bytes from slave2 (192.168.1.12): icmp_seq=1 ttl=64 time=0.488 ms
64 bytes from slave2 (192.168.1.12): icmp_seq=2 ttl=64 time=0.620 ms
64 bytes from slave2 (192.168.1.12): icmp_seq=3 ttl=64 time=0.610 ms
64 bytes from slave2 (192.168.1.12): icmp_seq=4 ttl=64 time=0.513 ms
64 bytes from slave2 (192.168.1.12): icmp_seq=5 ttl=64 time=0.753 ms

--- slave2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4083ms
rtt min/avg/max/mdev = 0.488/0.596/0.753/0.093 ms
root@master:/home/parallels#

```

图 4: 测试虚拟机集群对互联网和内网虚拟机之间的访问性

6 ssh 实现无密码访问

7 rsh 实现无密码访问

8 title