Linux子网搭建

首先给每台机器配置静态IP

端口名

注意要选对端口。第一个是我们需要的网线接口，第二个是内循环接口，第三个是wifi接口。

查看 ifconfig

```
hetao@hetao-XPS:~/Projects/JuliaProj/IBM-Julia/Source$ ifconfig
enx000ec6fa7adc: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.88  netmask 255.255.255.0  broadcast 192.168.0.188
        inet6 fe80::d1a1:ac3d:7015:9c04  prefixlen 64  scopeid 0x20<link>
        ether 00:0e:c6:fa:7a:dc  txqueuelen 1000  (Ethernet)
        RX packets 612  bytes 121910 (121.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3714  bytes 230384 (230.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 93146  bytes 9724518 (9.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 93146  bytes 9724518 (9.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp59s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.62.133.129  netmask 255.255.224.0  broadcast 10.62.159.255
        inet6 2001:da8:204:2512::1:ccfe  prefixlen 128  scopeid 0x0<global>
        inet6 fe80::d094:9ab0:4596:ed21  prefixlen 64  scopeid 0x20<link>
        ether 66:9d:05:f9:97:bc  txqueuelen 1000  (Ethernet)
        RX packets 5828725  bytes 2560715149 (2.5 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 668285  bytes 83805474 (83.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

设置

sudo ifconfig 端口名 192.168.0.X netmask 255.255.255.0 broadcast 192.168.0.255

设定c5~c8的IP为192.168.0.5~8
broadcast均为192.168.0.255

Linux子网搭建

注意各机器之间不是直接相连，而是通过路由相连。由于我们的集群的路由器IP被设为0.0.0.0，所以尝试ping 0.0.0.0
只要能ping通就行了。否则用route -n命令检查是否存在默认网关。如果没有，就设置一下：
sudo route add default gw 0.0.0.0
（注意到c5没有default route也能ping通0.0.0.0，显然是因为c5有网关为0.0.0.0的其他route可以ping。
另外注意route -n和route命令的区别。）

最后，刷新网络配置：
sudo /etc/init.d/network-manager restart
重启网络服务：
sudo service network-manager restart

到此为止，应该就能实现c5~c8之间互相ping通。

Linux ssh

ubuntu自带ssh-client，但没有ssh server。要下载openssh-server和openssh-sftp-server并安装。
下载地址为：
https://packages.ubuntu.com/disco/openssh-server
https://packages.ubuntu.com/disco/openssh-sftp-server

以下讲解如何从c5免密码远程登录c7。
在c5上的操作：首先su root，设定密码，生成一个root账户。
在c5上创建~/.ssh文件夹，进入并创建秘钥
ssh-keygen -t rsa
（连续按三个回车，不要输入自定义名称。）
在~/.ssh里新建一个config文件，输入

```
Host c7
HostName 192.168.0.7
User root
IdentityFile ~/.ssh/id_rsa
```

最后sudo service ssh restart重启ssh

Linux ssh

在c7上的操作：首先su root，设定密码，生成一个root账户。
在c7也创建一个~/.ssh。把c5的id_rsa复制过去，在那里再创建一个id_rsa副本并改名为authorized_keys
把authorized_keys复制到/root文件夹里
修改读取权限

```
chown -R c7:c7 .ssh
chmod 700 .ssh
chmod 600 .ssh/authorized_keys
```

打开文件sudo vi etc/ssh/sshd_config，修改键值如下：

```
PermitRootLogin yes
PubkeyAuthentication yes
PasswordAuthentication no
AuthorizedKeysFile     .ssh/authorized_keys  /root/authorized_keys
```

（注意要删掉前面的#）
最后sudo service sshd restart重启sshd
（ssh指代ssh client，sshd指代ssh server）

最后回到c5，输入ssh c7@192.168.0.7或ssh root@192.168.0.7登录。

Julia配置

在c5上做如下设置：
把Julia文件放在合适位置，如~/Softwares/Julia-1.2.0
添加环境变量
sudo vi /etc/bash.bashrc
写入 export PATH=~/Softwares/Julia-1.2.0/bin/:$PATH
刷新 source /etc/profile
用env检查PATH变量是否已成功加入julia路径
（不要直接修改profile，否则每次启动新的terminal都会必须手动刷新/etc/profile）
这样确保任意位置都能启动julia

在c7上创建一个/home/c5/Softwares文件夹，把julia-1.2.0复制过去。
这是因为c5远程调用c7时，会默认在与c5一致的路径下查找julia可执行文件。
同理，在c7上也要创建一个/home/c5/Projects/JuliaProj/ClusterDemo，也就是与c5一致的代码路径。
如果没有这两步，就会出现“can't cd to ……”的错误。

可以考虑用共享文件系统NFS解决路径问题。

Julia配置

在c5上启动并行程序的两种方法：

方法一：
先启动julia REPL，
用addprocs()命令添加进程，格式为：
addprocs([(“c7@192.168.0.7, 8)])
注意addprocs()的参数是一个Vector，其中元素是元组，每个元组又包含一个字符串和一个整数。字符串表示远程账户，整数表示在远程主机上发起的线程数。
可以把这个命令写入一个文件并在启动julia时加载，例如startupfile.jl。
那么启动命令为：julia -L startupfile.jl

方法二：
编写一个machinefile文件，分行写入远程账户：
例如：c7@192.168.0.7
每个账户单独一行
然后julia --machine-file machinefile启动REPL，就会自动发起多个进程。
尽管官方文档说machinefile默认发起的进程数等于逻辑线程总数，但我实测发现只启动了两个，一个在c5一个在c7。可见machinefile并不是很靠谱。
可以再用addprocs()手动添加进程。

一个startupfile.jl示例

```
using Distributed
# 本地主机发起8个进程（包含主进程）
addprocs(7)
# 远程主机发起8个进程
addprocs([(“c7@192.168.0.7, 8)])
```