# 10步搭建Ubuntu 18.04 Linux计算集群

如图1，搭建一个采用Server-Client架构的Linux计算集群，Server是一台提交并行计算任务的PC，Client是称为计算节点的若干台PC。



图1 Ubuntu Linux集群示意图

## 第1步：安装Ubuntu 18.04 OS.

Desktop or Server Edition?

## 第2步：连接数据交换机(SWITCH)

将2台（或更多）电脑连接到数据交换机。使用高速网线（或光缆），Ethernet网线有不同的最大带宽。

## 第3步：设置网络地址

使用Ubuntu GUI分配IP地址

Setting->NetWork

然后，检查分配的IP地址：ifconfig –a

可能有有线和无线的IP

10.0.0.1 on mini system （Server）

10.0.0.2 in Xeon system （Client）

-------------------------------------------------------------------

可以使用Ubuntu的NetworkManager的nmcli查看网络连接情况：

nmcli device status

以太网或WIFI

设置共享连接：

nmcli connection add type ethernet ifname enp2s0 ipv4.method shared con-name cluster

以上操作增加了防火墙准则来管理cluster网络内的数据通信，启动DHCP服务器

ifconfig，出现10.42.0.1和cluster网络中的另一台电脑的动态IP：10.42.0.1、24

## 第4步：安装SSH

在每台电脑上都安装SSH：apt-get install openssh-server

SSH配置的细节…?

查看SSH和SSHD(服务器Deamon)的配置选项：

gedit /etc/ssh/ssh\_config

gedit /etc/ssh/sshd\_config

确保每个计算节点运行SSHD，命令：systemctl status sshd

## 第5步：设置无密码访问

启用无密码的SSH访问，运行：ssh-keygen –t rsa –b 4096

接受所有默认设置。将生成2个文件：~/.ssh/id\_rsa和~/.ssh/id\_rsa.pub，后一个文件就是public key，该文件内容需要拷贝到远程机器上的~/.ssh/authorized-keys

查看public key内容：cat ~/.ssh/id\_rsa.pub

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDSg3DUv2O8mvUIhta2J6aoXyq7lQ9Ld0Ez1exOlM+OGONH...cvzQ== user**@**mini

然后，登录计算节点：ssh 10.0.0.2

gedit ~/.ssh/authorized-keys

粘贴上面的public key内容到文件尾部，保存。

对于大型集群，可以对所有计算节点(Client)都使用相同的private key，可以复制authorized-keys到所有计算节点：

scp ~/.ssh/authorized\_keys 10.0.0.2:~/.ssh/

(scp是采用SSH复制文件的命令)

最后，添加host names到/etc/hosts文件：

more /etc/hosts

127.0.0.1 localhost

127.0.1.1 mini

10.0.0.2 xeon

可以测试一下能够远程登录计算节点： ssh xeon

登录就无需输入密码了！

## 第6步：安装MPI

可以安装默认的OpenMPI: apt-get install libopenmpi-dev

或者，自己编译安装。

## 第7步：测试MPI

mpic++

mpirun –np 2 hostname # 确保能在主机上运行mpirun

mpirun –np 2 –host 10.0.0.2:2 hostname # 确保可以通过网络接口启动进程

(上述是openmpi的语法，MPICH使用不同的参数，要查看手册)

在主机上运行hostname，使用IP地址10.0.0.2，该系统有2个计算插槽(:2)：2核？

还可以在xeon计算节点上启动2份hostname，运行：

mpirun -np 3 -host 10.0.0.2:2 -host 10.0.0.1 **hostname**

输出：

mini

xeon

xeon

表示：可以在多个系统上启动MPI作业，现在有了基本的集群能力。

假设已经在/etc/hosts中添加了远程IP地址，上面的命令还可以是：

mpirun -np 3 -host xeon:2 -host mini **hostname**

需要在命令中指定主机名很麻烦，可以创建hostfile

more ~/hosts

10.0.0.2 slots=10

10.0.0.1 slots=6

这样就表明：Xeon系统有10个CPU核心，个人工作站有6个CPU核心。这样，模拟进程在移到"mini"系统之前，首先利用新的Xeon系统。

mpirun -np 11 -hostfile ~**/**hosts **hostname**

输出：

mini

xeon

xeon

xeon

xeon

xeon

xeon

xeon

xeon

xeon

xeon

对于openmpi如果启动的进程数大于设置的16个，会报错：

mpirun -np 20 -hostfile ~**/**hosts **hostname**

There are not enough slots available **in** the system to satisfy the 20 slots

that were requested by the application:

**hostname**

但MPICH不会，默认再利用可获取的计算资源。可以将openmpi设置为类似的行为，使用oversubscribe参数：

mpirun -np 20 -hostfile ~**/**hosts -oversubscribe **hostname**

## 第8步：设置网络文件系统(NFS)

自此都是使用hostname命令，这对所有系统都是默认可获取的。

使用MPI，本质上仅使用网络连接来允许多个运行作业之间的通信。各作业运行在自己的电脑上，访问自己的硬件驱动。这就要求所有的计算程序要在所有计算节点上位于相同路径！

例如，在一个计算节点上运行：mpirun –np 3 ./mpi\_test 没问题

但是，远程运行测试程序，会出错！

mpirun -np 3 -host xeon:3 .**/**mpi\_test # 出错！

因为在Xeon的home路径下没有mpi\_test可执行程序！可以使用scp或rsync拷贝：

rsync -rtv .**/**mpi\_test xeon:

mpirun -np 3 -host xeon:3 .**/**mpi\_test # 没问题了！

这很不方便！

解决办法就是：设置一个网络驱动！这在Linux系统上很方便，参考：

https://www.digitalocean.com/community/tutorials/how-to-set-up-an-nfs-mount-on-ubuntu-16-04

首先，需要安装NFS核心和通用程序：

apt **install** nfs-kernel-server nfs-common

然后，创建共享路径的挂载点，共享/home/lijian

在mini上创建软连接到上述路径：**ln** -s **/**nfs **/**home**/**user

ls –la /nfs

然后，添加如下连接到/etc/exports

**more /**etc**/**exports

添加 **/**home**/**user xeon**(**rw**)**

该命令允许xeon可以read-write访问某指定文件夹。

然后，在远程Client端(xeon)上创建/nfs挂载点。更新/etc/fsatb包含：

user@xeon:~$ **sudo mkdir /**nfs

user@xeon:~$ **more /**etc**/**fstab

...

10.0.0.1:**/**home**/**user **/**nfs nfs defaults 0 0

运行： user@xeon:~$ **sudo mount** –a 启动fstab

配置完成后，可以到文件夹下，查看Server (mini)上home路径下的内容：

user@xeon:~$ **cd /**nfs

user**@**xeon:**/**nfs$ **ls** -la

total 60084

。。。

在/etc/passwd中设置User ID，然后更新/etc/group

user**@**xeon:**/**nfs$ **more /**etc**/passwd**

...

user:x:1000:1000:User,,,:**/**home**/**user:**/**bin**/bash**

user**@**xeon:**/**nfs$ **more /**etc**/**group

...

user:x:1000:

最后，看看是否能工作。到Server的/nfs路径下，试着运行程序：

user**@**mini:**/**nfs$ mpirun -np 3 -host xeon:2 -host mini:1 .**/**mpi\_test

## 第9步：启用防火墙（可选）

Ubuntu使用ufw命令控制防火墙。默认防火墙是关闭的。

使用如下命令启用防火墙：ufw enable

现在运行并行计算代码，但没有输出，MPI超时：

user**@**mini:**/**nfs$ mpirun -np 3 -host xeon:2 -host mini:1 .**/**mpi\_test

--------------------------------------------------------------------------------------

A process or daemon was unable to **complete** a TCP connection

to another process:

Local host: xeon

Remote host: 192.168.1.4

This is usually caused by a firewall on the remote host. Please

check that any firewall **(**e.g., iptables**)** has been disabled and

try again.

---------------------------------------------------------------------------------

可以对防火墙打开一个洞，允许任何系统位于10.0.xxx.xxx的子网络，使用：

user**@**mini:**/**nfs$ **sudo** ufw allow from 10.0.0.0**/**16

user**@**mini:**/**nfs$ mpirun -np 3 -host xeon:2 -host mini:1 .**/**mpi\_test

I am 2 of 3 on mini

I am 0 of 3 on xeon

I am 1 of 3 on xeon

## 第10步：执行计算

现在有了MPI集群，可以执行并行计算了!

## 计算节点的温度监控

lm-sensors

sudo sensors-detect

watch sensors