

## 1. MPI 集群搭建

本组实现的 MPI 集群搭建是两台计算机在同一个局域网下搭建的。如果没有公共的在同一个局域网下，可以通过内网穿透然后再进行连接，或者使用公共的服务器也可以解决这个问题，本次使用 deepin 系列的计算机测试，命令各趋向于 ubuntu。

在搭建集群之前，首先要保证，两台计算机已经安装了 MPICH 并且版本最好一致。此外除了 MPICH 版本一致外，GCC 以及 fortran 的版本最好也要相同，防止不能运行共同的文件，最好的方法就是使用两个一样 Linux 版本的计算机测试。在测试之前，需要保证两个计算机的用户名是相同的，不相同最好改一下用户名。下面是具体的安装方法。

### 1) 两台计算机安装 SSH，并设置 SSH 免密登录

使用 `ifconfig` 命令查看两个计算机的 IP 地址，修改两个计算机的 `hosts` 文件，在两个计算机的 `hosts` 文件分别添加节点 IP。

```
sudo vim /etc/hosts
```

```
192.168.1.3 node1
```

```
192.168.1.7 node2
```

可以通过 `ping` 命令判断是否机器可以 `ping` 到，出现错误要检查 `hosts` 文件是否修改正确。

```
ping node1
```

```
ping node2
```

各计算机安装 `ssh`。

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

尝试是否可以通过 `ssh` 命令进行各计算机的远程登录，此时的登录是需要密码的。

```
ssh 用户名@node1
```

```
ssh 用户名@node2
```

接下来需要生成 `ssh` 的公钥和私钥，并讲两个节点的公钥都添加到自己的 `authorized_keys` 中，这样就可以实现免密登录了。具体操作如下：

```
ssh-keygen -t rsa 两个计算机都要操作，生成钥匙在.ssh 文件中
```

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys 两个计算机都要操作，对自己计算机的认证，该操作后使用 ssh 可以对自己免密登录
```

```
scp ~/.ssh/id_rsa.pub 用户名@node2:~/.ssh/id_rsa.pub.node1 node1 节点计算机需要的操作，将 node1 的公钥传到 node2
```

```
cat ~/.ssh/id_rsa.pub.node1 >> ~/.ssh/authorized_keys node2 节点计算机需要的操作，将刚刚传到的 node1 公钥添加认证
```

```
scp ~/.ssh/id_rsa.pub 用户名@node1:~/.ssh/id_rsa.pub.node2 node2 节点计算机需要的
```

操作，将 node1 的公钥传到 node1

`cat ~/.ssh/id_rsa.pub.node2 >> ~/.ssh/authorized_keys` node1 节点计算机需要的操作，将刚刚传到的 node2 公钥添加认证

再次使用 ssh 命令测试是否 node1 和 node2 之间都可以进行免密登录。

`ssh 用户名@node1`

`ssh 用户名@node2`

如果可以相互连接成功，表示 node1 和 node2 之间通过 ssh 可以进行免密登录，之后就可以尝试集群共同运行程序进行尝试，在测试之前，首先要在两个计算机上创建一个系统的目录。两个计算机都创建 /home/mpi\_share 文件夹，然后使用 MPI 测试的 cpi 程序进行测试。两个计算机都要在 /home/mpi\_share 文件夹中加入 cpi 程序。然后再同加入 mpi\_config 文件，作为运行的配置文件，内容如下。

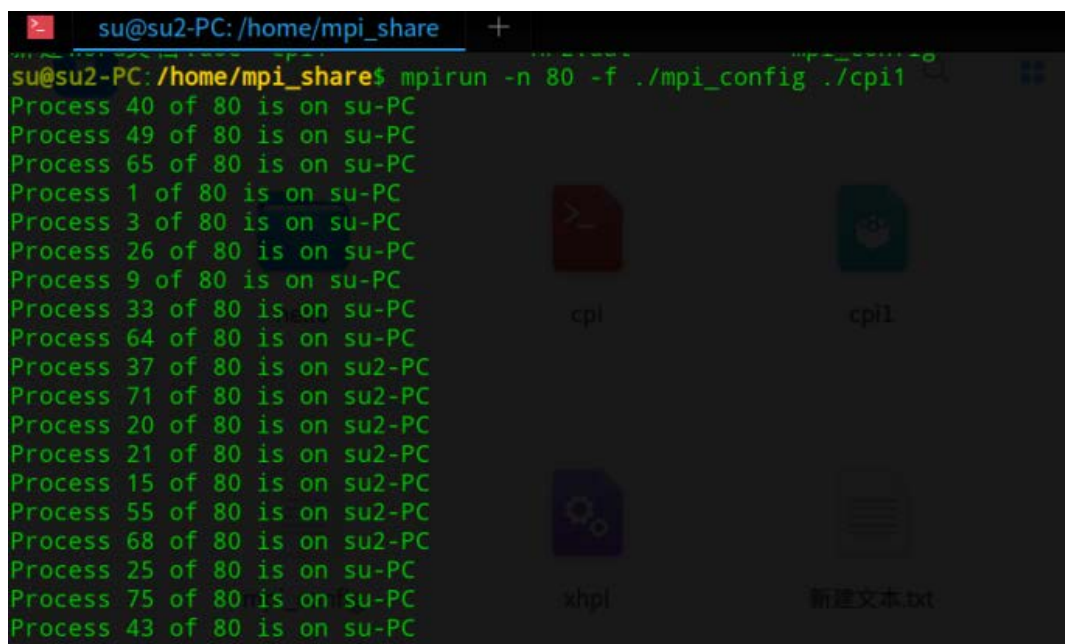
node1:4

node2:4

这里的添加根据自己的 CPU 核心数进行添加，然后在任意计算机进行运行。

`mpirun -n 80 -f ./mpi_config ./cpi1`

运行结果如下

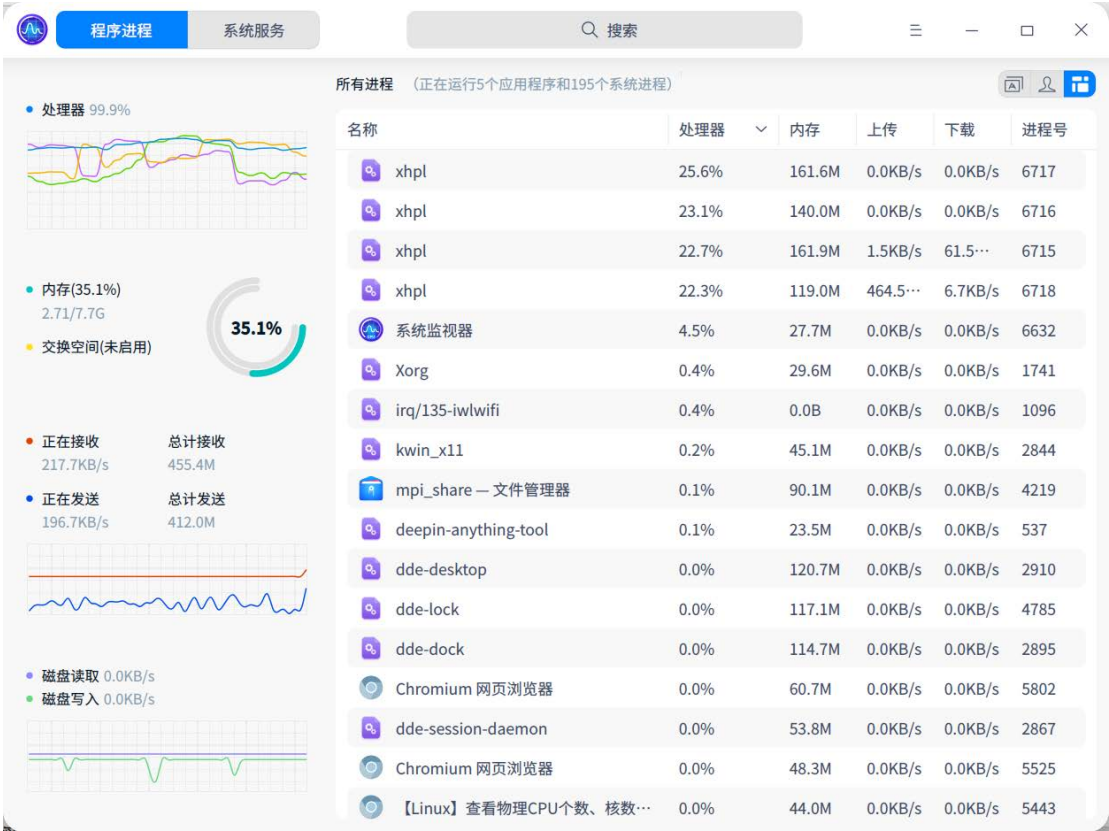
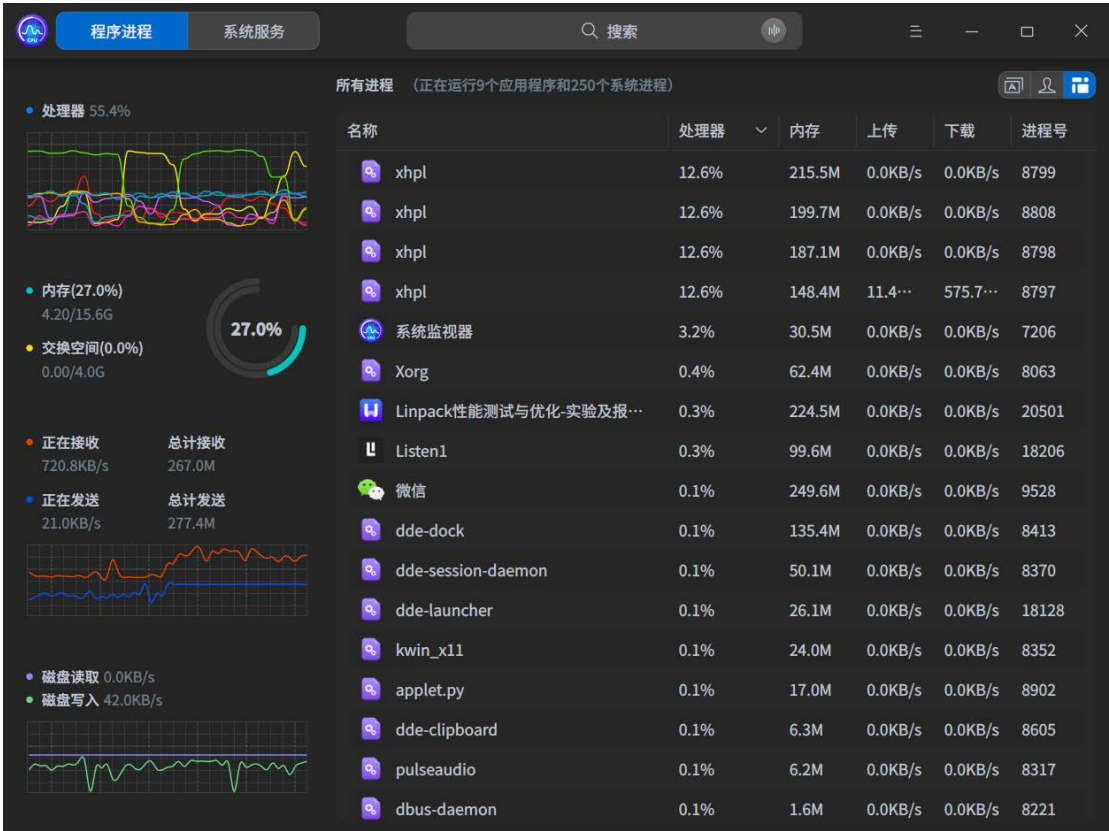
A terminal window titled 'su@su2-PC: /home/mpi\_share' showing the output of the command 'mpirun -n 80 -f ./mpi\_config ./cpi1'. The output lists 80 processes and their locations. Processes 40, 49, 65, 1, 3, 26, 9, 33, 64, 37, 71, 20, 21, 15, 55, 68, 25, 75, and 43 are on 'su-PC'. Processes 2, 5, 8, 11, 14, 17, 22, 23, 24, 27, 28, 30, 31, 32, 34, 35, 36, 38, 39, 41, 42, 44, 45, 46, 47, 48, 50, 51, 52, 53, 54, 56, 57, 58, 59, 60, 61, 62, 63, 66, 67, 69, 70, 72, 73, 74, 76, 77, 78, 79, and 80 are on 'su2-PC'. On the right side of the terminal window, there are icons for files named 'cpi', 'cpi1', 'xhpl', and '新建文本.txt'.

如果运行结果和上述相同，表示可以在集群下使用 MPI 运行文件，同理也可以将 HPL.dat 和 xhpl 文件都添加到该目录下，然后任意计算机运行测试集群的性能。

`mpirun -np 8 -f ./mpi_config ./xhpl`

在进行测试时，两台计算机 CPU 和内存资源都会被一定量的占用，和两台计算机单独运行的表现差异不大，但是需要占用相对较多的网络资源。在同一个局域网下，其他设备的

网络速度会降低很多。下图分别为运行情况下，node1 和 node2 资源使用情况。



初步测试的结果是在集群上进行运行，效果不如 node1 进行单机运行，初步考虑可能是由于网络的问题导致运行速度不增反减。

在测试运行时，有一个问题就是每次需要执行文件时都需要在两个计算机上都将同一个节点放到同一个目录下，这个操作是比较繁琐的，可以通过安装 NFS 使得两个文件夹进行共享目录，在该目录下任意节点进行添加或者删除文件，其他的计算机的目录下都会响应进行变化。首先每个计算机节点安装 NFS。

```
sudo apt-get install nfs-kernel-server
```

在两个节点中要选取一个作为服务器，其他的做客户端，在这里使用的是 node1 作为服务器，node2 作为客户端。node1 需要修改 exports 文件。

```
sudo vim /etc/exports
```

```
/home/mpi_share 192.168.1.3(rw,sync,no_root_squash,no_subtree_check)
```

```
/home/mpi_share 192.168.1.3(rw,sync,no_root_squash,no_subtree_check)
```

```
sudo /etc/init.d/nfs-kernel-server restart
```

在其他节点中需要进行挂载，在 node2 中运行

```
sudo mount -t nfs node1:/home/mpi_share /home/mpi/share
```

然后可以在/home/mpi/share 文件夹中添加文件观察是否可以共享