

分布式存储编程—MPI 实验 参考教程
V0.4

2020

华南理工大学 计算机学院

目录

第1章	Windows 下MPI 配置	3
1.1	配置 Microsoft MPI	3
1.1.1	Microsoft MPI 安装	3
1.1.2	Visual Studio 的编译设置	6
1.1.3	使用 Microsoft MPI 单机运行 MPI 程序	12
1.2	配置 MPICH	13
1.2.1	MPICH 安装	13
1.2.2	Visual Studio 的编译设置	13
1.2.3	使用 MPICH 单机运行 MPI 程序	14
第2章	Linux 下配置MPI (以MPICH 为例)	16
2.1	MPI 安装	16
2.2	编译与运行 (单节点)	17
第3章	** (选) 多节点运行 (集群环境)	18
3.1	配置 SSH	18
3.1.1	配置 ssh 无密码登陆	18
3.1.2	配置 ssh 昵称	19
3.2	执行程序	19
第4章	**(选)虚拟机实验 (以Linux 为例)	20
4.1	环境配置	20
4.2	新建虚拟机	21
4.3	配置集群网络	24
4.3.1	已安装多个虚拟机情况:	24
4.3.2	安装单个虚拟机需要复制情况:	27
4.4	配置 SSH (与 3.1 重复)	32
4.4.1	配置 ssh 无密码登陆	33
4.4.2	配置 ssh 昵称	34

第 1 章 Windows 下 MPI 配置

MPI 是一个接口标准，有各种不同的实现方式。本章分别描述了在 Windows 下两种 MPI 实现的配置方法。1.1 节展示了 Microsoft MPI 的配置步骤，1.2 节描述了 MPICH 的配置步骤。在 Windows 下进行 MPI 程序开发，仅需选一种进行配置即可。1.2 节描述的 MPICH 版本较低，推荐使用 Microsoft MPI 按照 1.1 节进行配置。

1.1 配置 Microsoft MPI

1.1.1 Microsoft MPI 安装

本节以 Microsoft MPI 为例进行配置。Microsoft MPI 的下载地址为：
<https://docs.microsoft.com/en-us/message-passing-interface/microsoft-mpi>
点击红框处进入下载页。

Microsoft MPI

03/28/2018 • 2 minutes to read • 3 👤

Microsoft MPI (MS-MPI) is a Microsoft implementation of the [Message Passing Interface standard](#) for developing and running parallel applications on the Windows platform.

MS-MPI offers several benefits:

- Ease of porting existing code that uses [MPICH](#).
- Security based on Active Directory Domain Services.
- High performance on the Windows operating system.
- Binary compatibility across different types of interconnectivity options.

MS-MPI Source Code

Microsoft MPI source code is available on [GitHub](#).

MS-MPI Downloads

The following are current downloads for MS-MPI:

- **MS-MPI v10.1.2 (new!)** see [Release notes](#)
- [Debugger for MS-MPI Applications with HPC Pack 2012 R2](#)

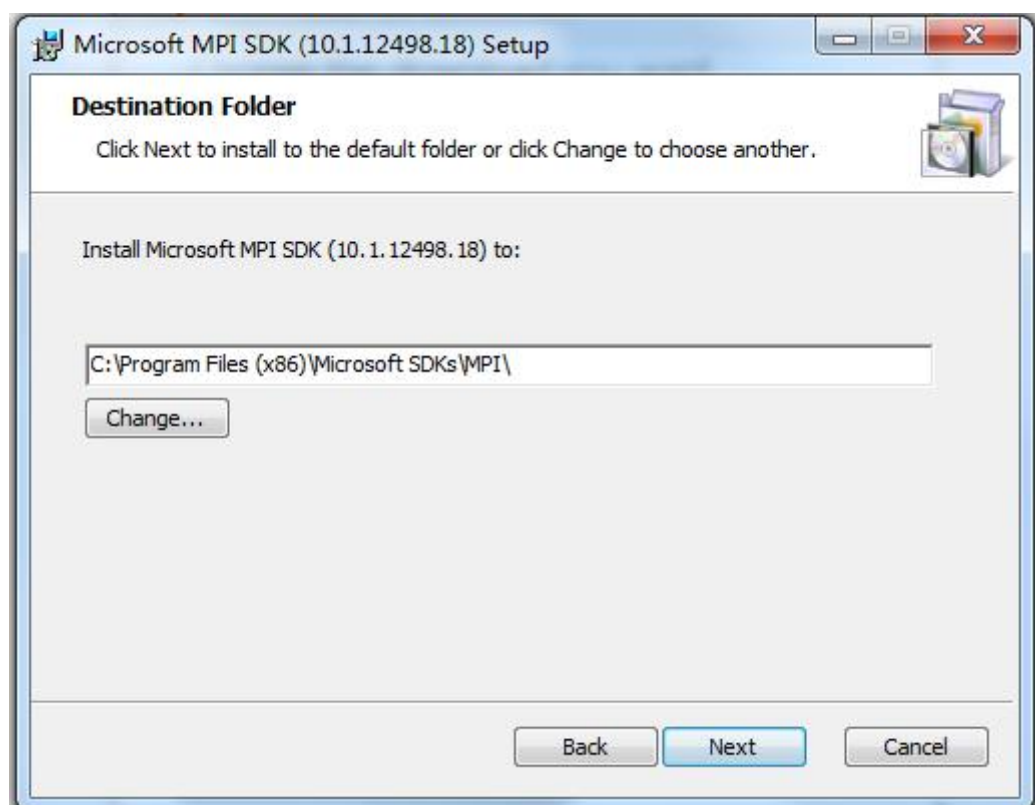
Earlier versions of MS-MPI are available from the [Microsoft Download Center](#).

注意，以上两个安装程序均需要下载安装。msmpisetup.exe 用于安装 Microsoft MPI 程序的运行环境。msmpisdk.msi 用于安装 Microsoft MPI 程序的

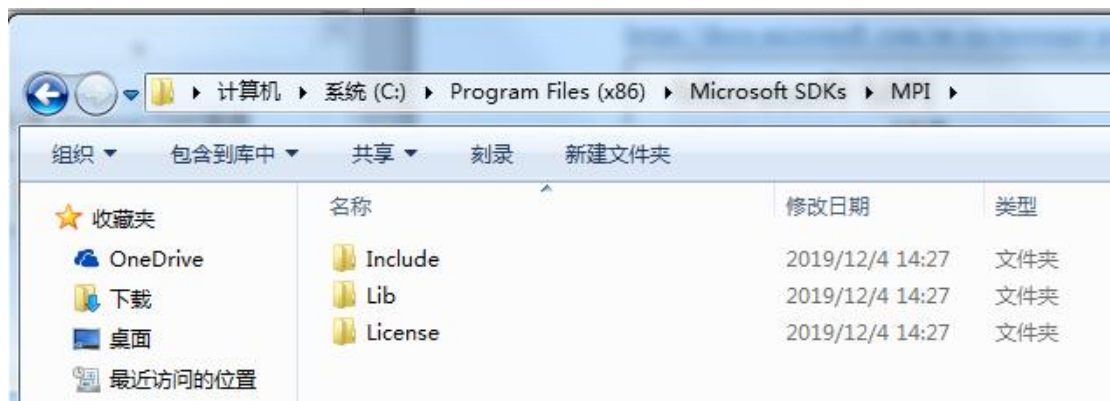
开发环境。



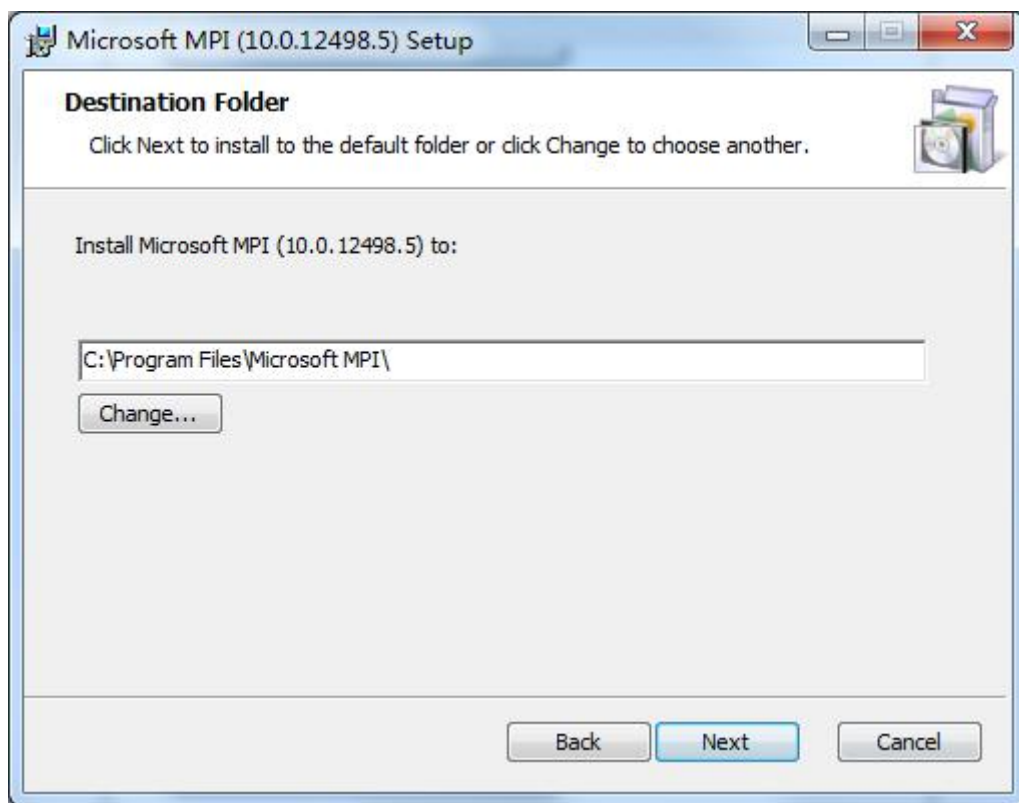
双击 msmpisdk.msi 安装 Microsoft MPI SDK，请记下其安装位置，后面的步骤需要用到。



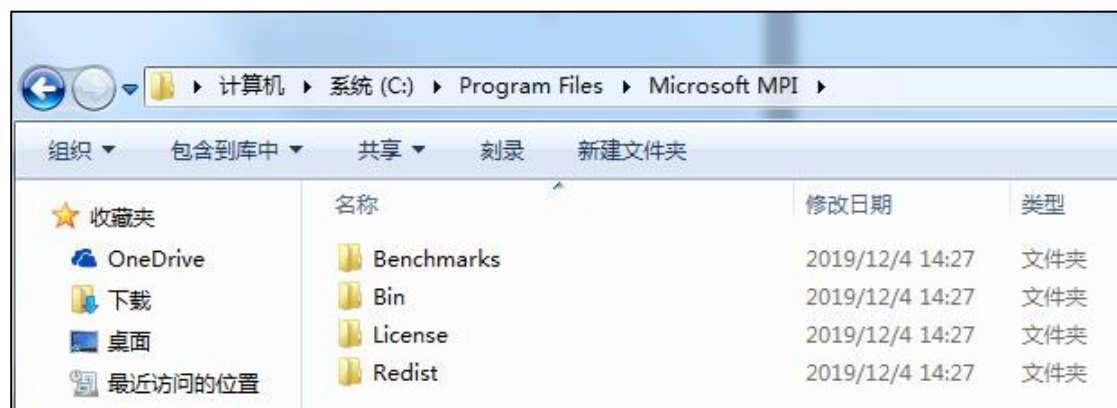
安装目录下的内容如图所示。



双击 msmpisetup.exe 安装 Microsoft MPI，请记住其安装位置，后面的步骤需要用到。

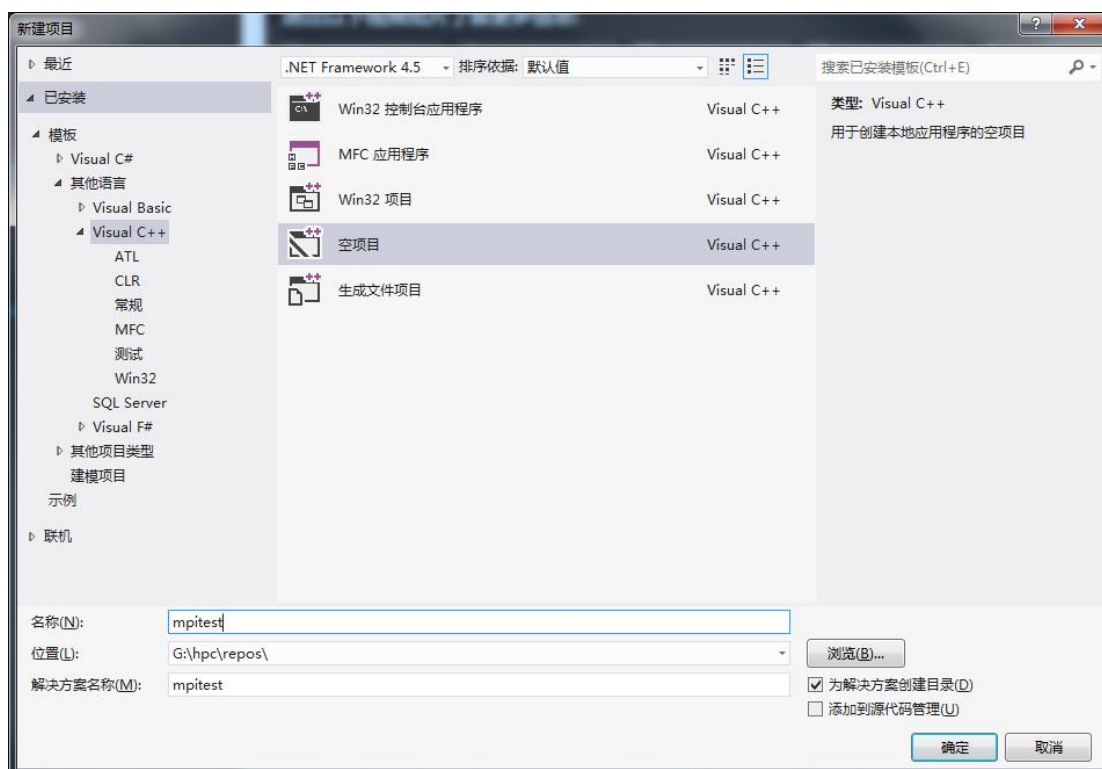


安装目录下的内容如图所示。

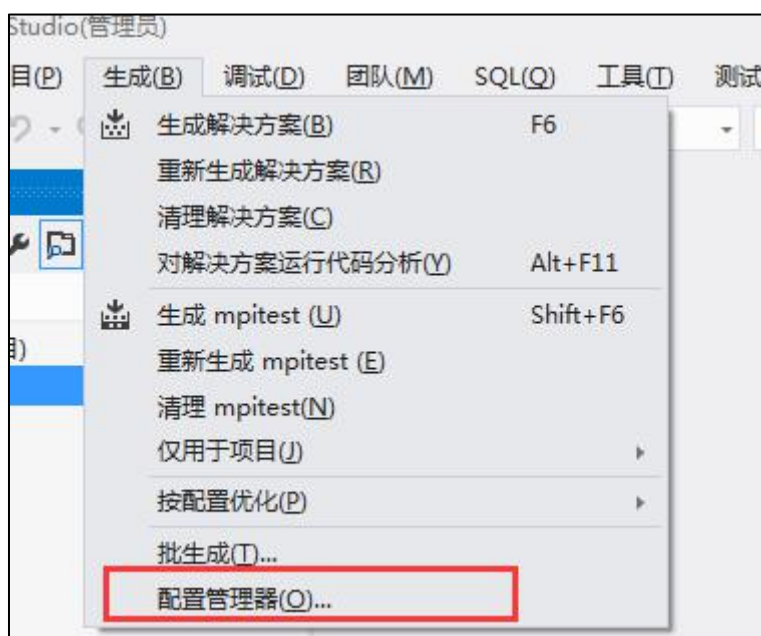


1.1.2 Visual Studio 的编译设置

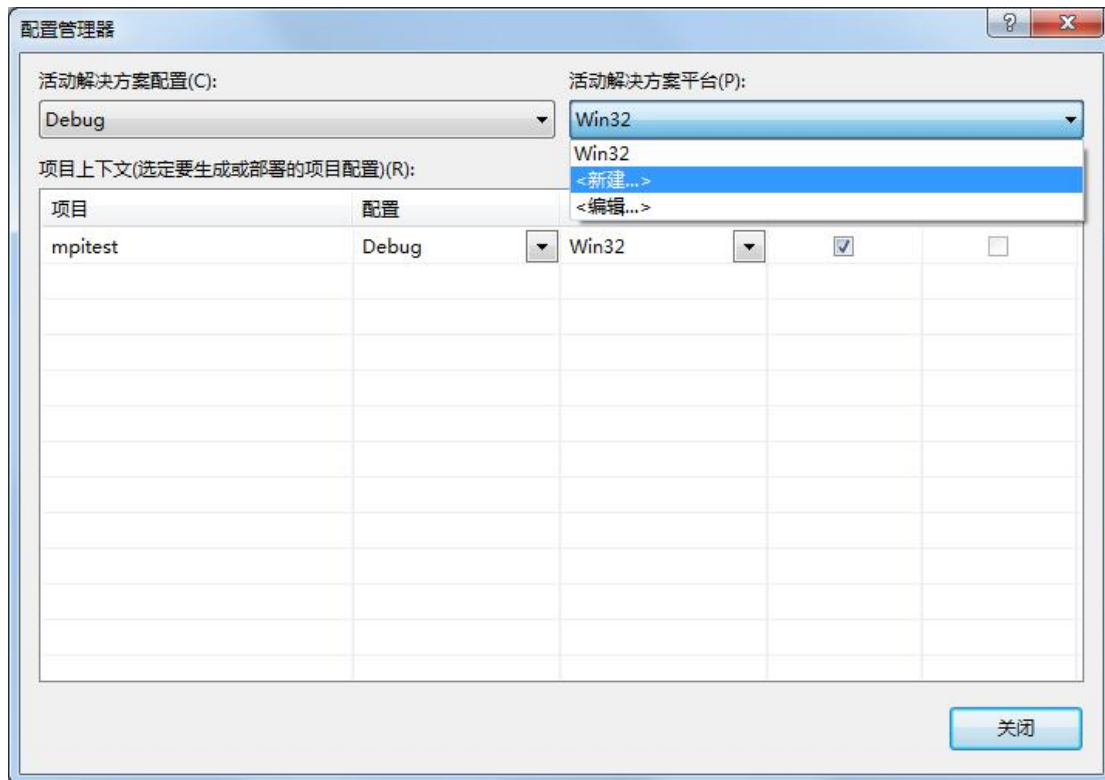
打开 Visual Studio 创建一个新的空项目。项目名称和位置可自由填写。



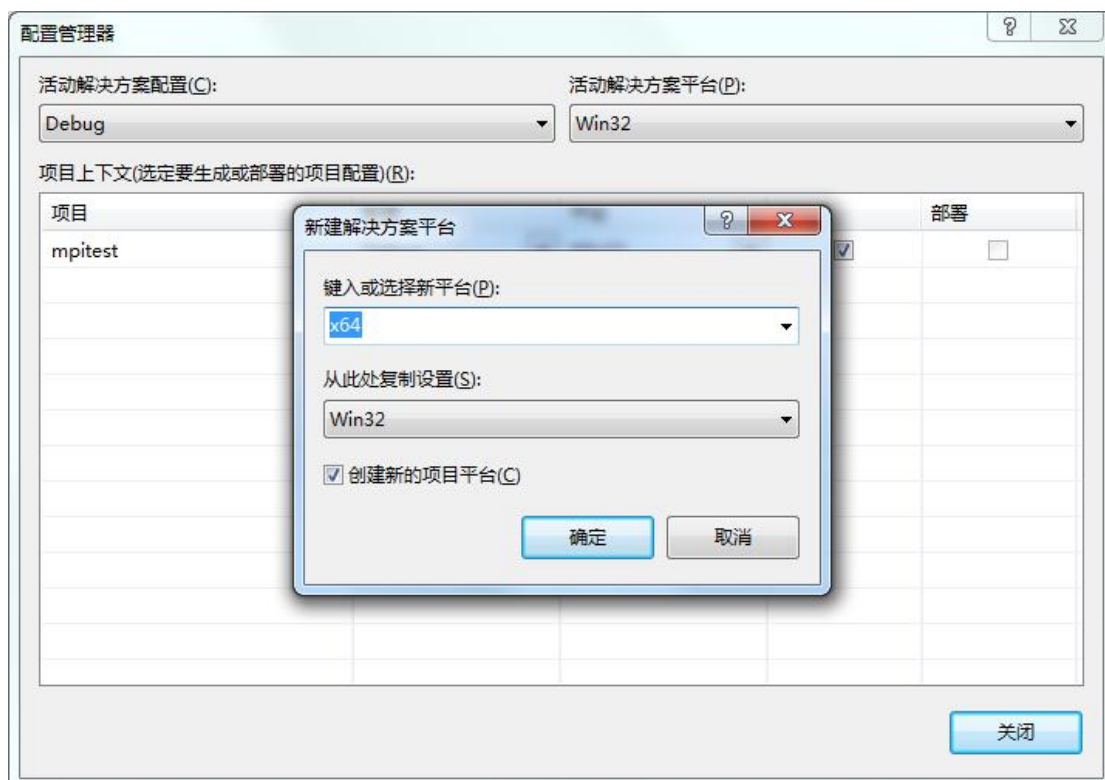
创建好项目之后，点击“生成” - “配置管理器”。



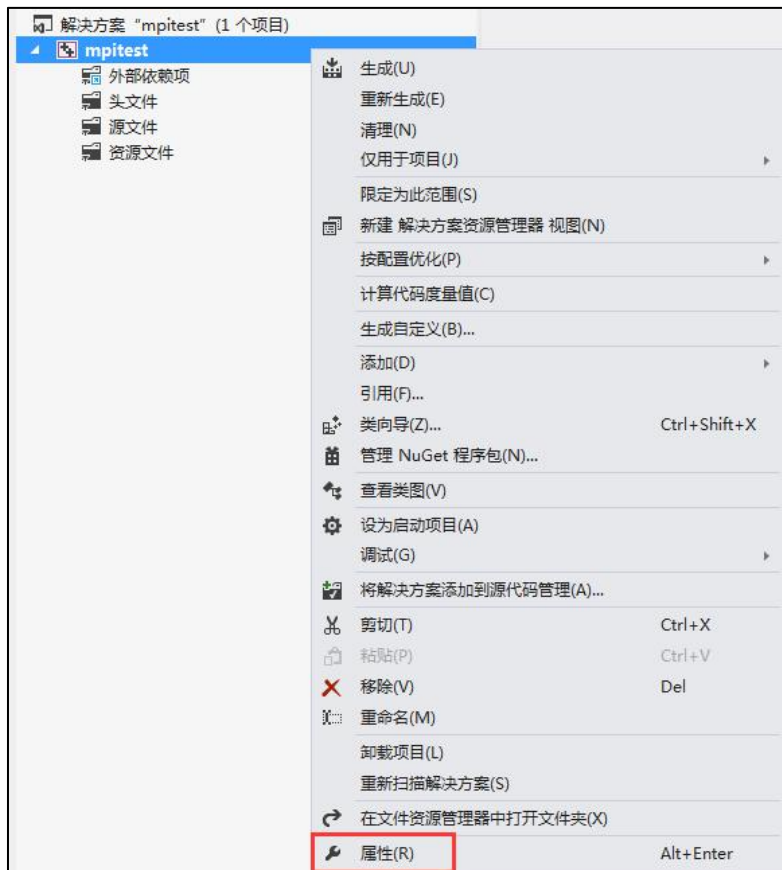
检查“活动解决方案平台”是否为“x64”，是的话则不用进行任何操作，否则点击“<新建...>”。



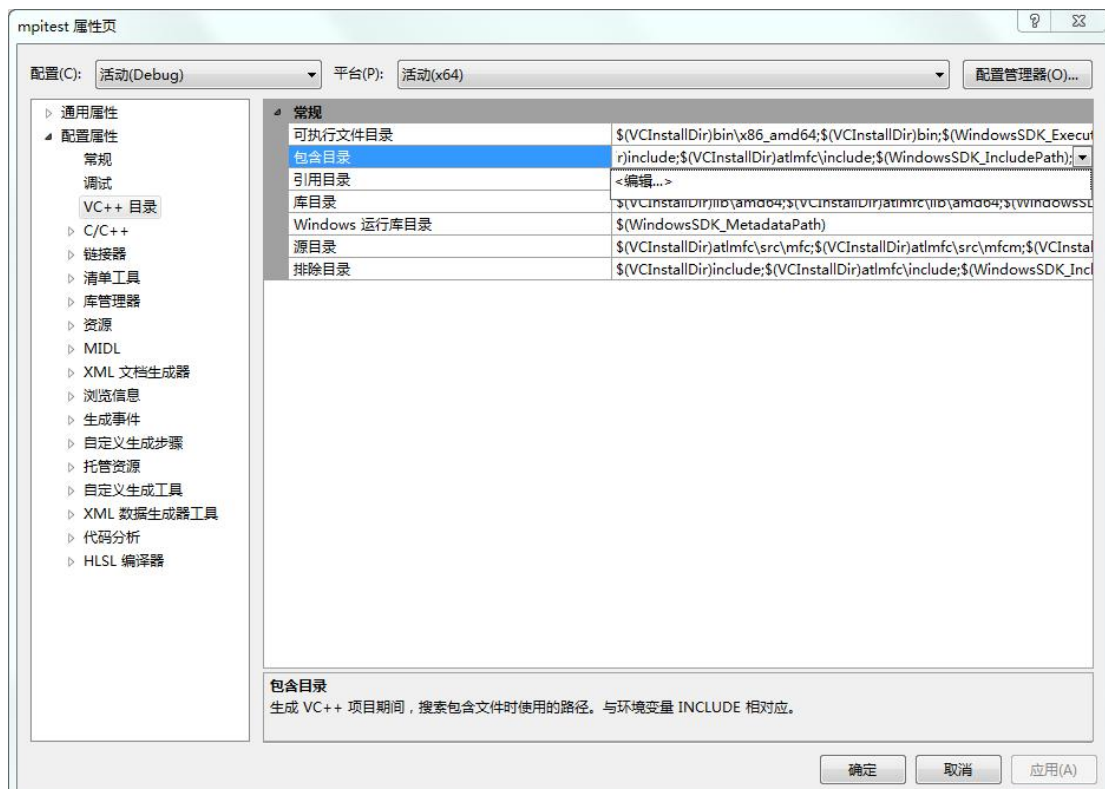
在“键入或选择新平台”中选择“x64”，然后确定。



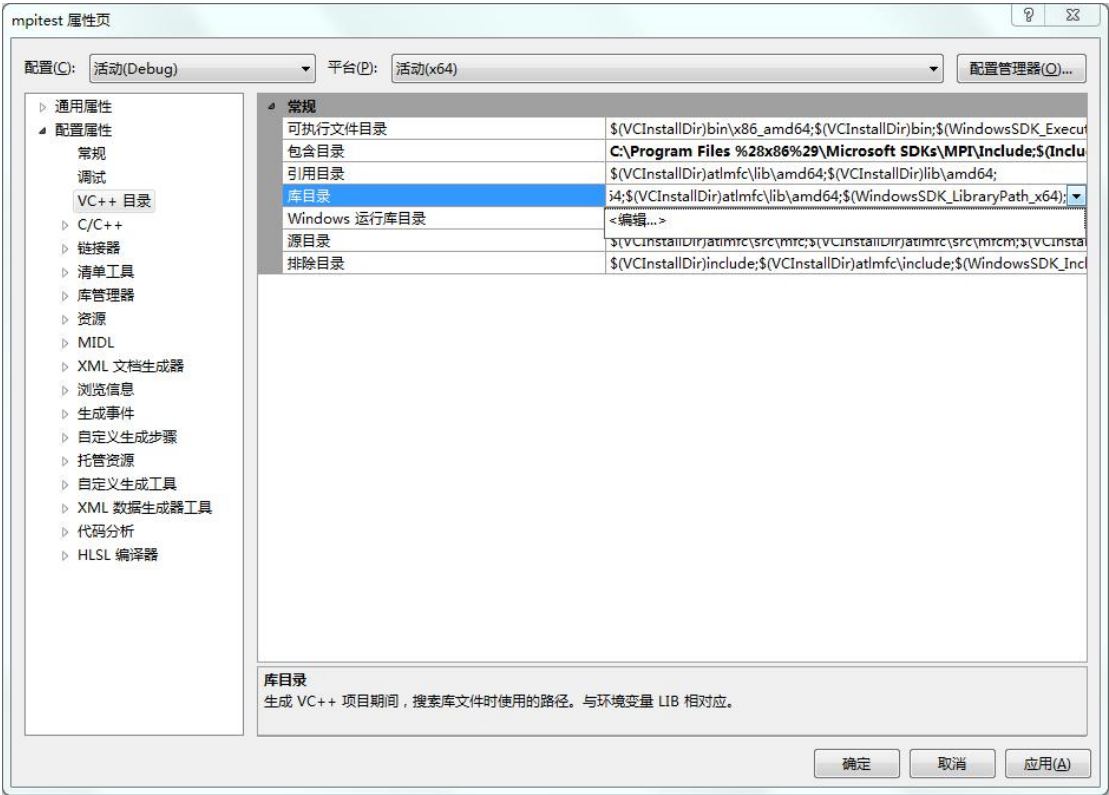
完成之后，右键点击当前项目，选择“属性”。



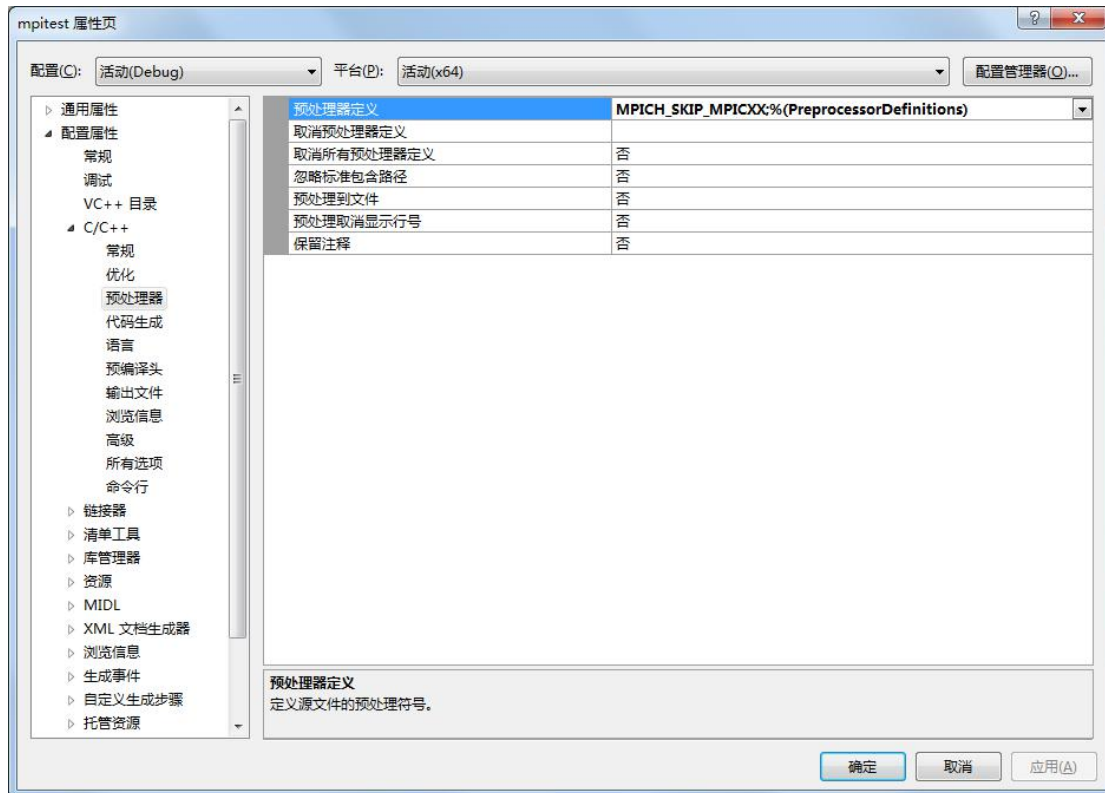
在左边一列选择“配置属性” - “VC++ 目录”，在“包含目录”中添加“C:\Program Files (x86)\Microsoft SDKs\MPI\Include”。注意，如果安装 Microsoft MPI SDK 时更改过安装位置，在这一步需要将路径修改为对应的安装位置。



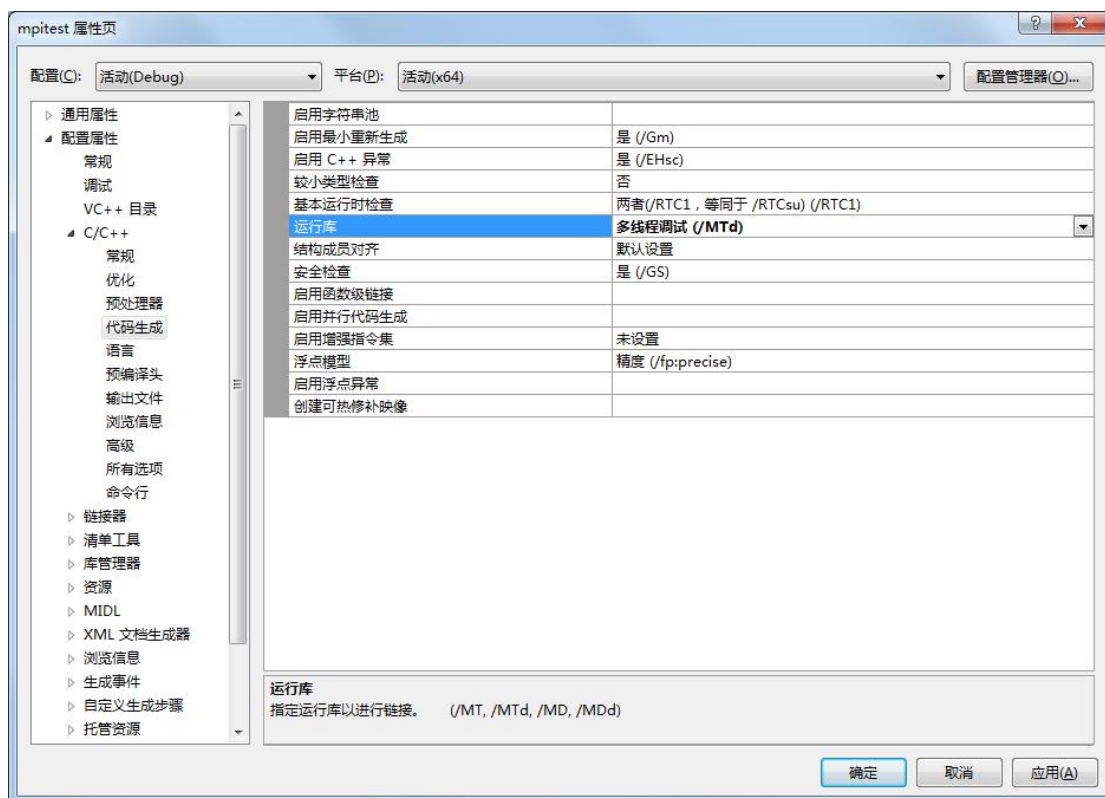
在“库目录”中添加“C:\Program Files (x86)\Microsoft SDKs\MPI\Lib\x64”。注意，如果安装 Microsoft MPI SDK 时更改过安装位置，在这一步需要将路径修改为对应的安装位置。



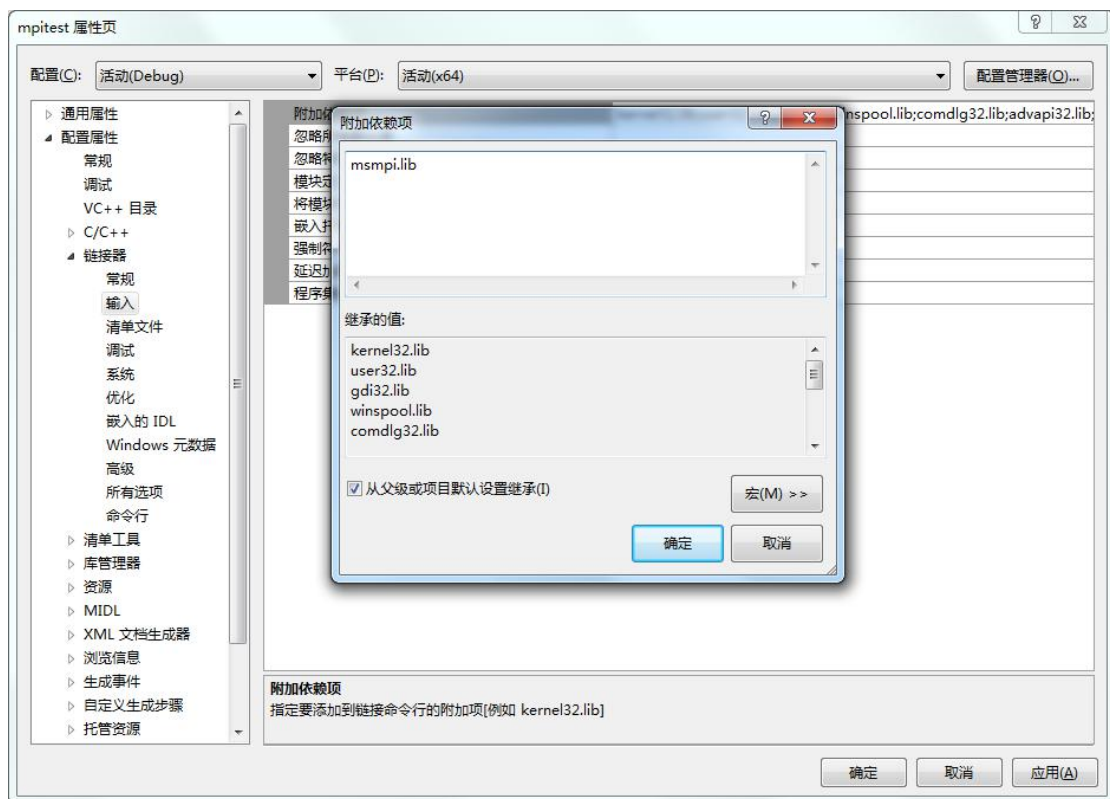
在左边一列中选择“配置属性” - “C/C++” - “预处理器”。在“预处理器定义”中添加“MPICH_SKIP_MPICXX”。



在左边一列中选择“配置属性” - “C/C++” - “代码生成”。在“运行库”中选择“多线程调试(/MTd)”。



在左边一列中选择“配置属性”-“链接器”-“输入”。在“附加依赖项”中添加“msmpi.lib”。



保存所有更改。新建一个“main.cpp”文件，输入以下内容并点击“生成”-“生成解决方案”进行编译。编译完成后会在项目目录下生成一个可执行文件，在本例中，该文件的位置是 G:\hpc\repos\mpitest\x64\Debug\mpitest.exe。

1.1.3 使用 Microsoft MPI 单机运行 MPI 程序

打开 Microsoft MPI 安装目录下的 Bin 文件夹，即 C:\Program Files\Microsoft MPI\Bin。注意，如果在安装 Microsoft MPI 时更改过安装位置，这一步需要修改为对应的安装位置。按住 Shift 键，右键点击空白处，点击“在此处打开命令窗口”。输入以下命令运行刚刚编译的 MPI 程序：mpexec.exe -n 10 G:\hpc\repos\mpitest\x64\Debug\mpitest.exe，结果如图所示。

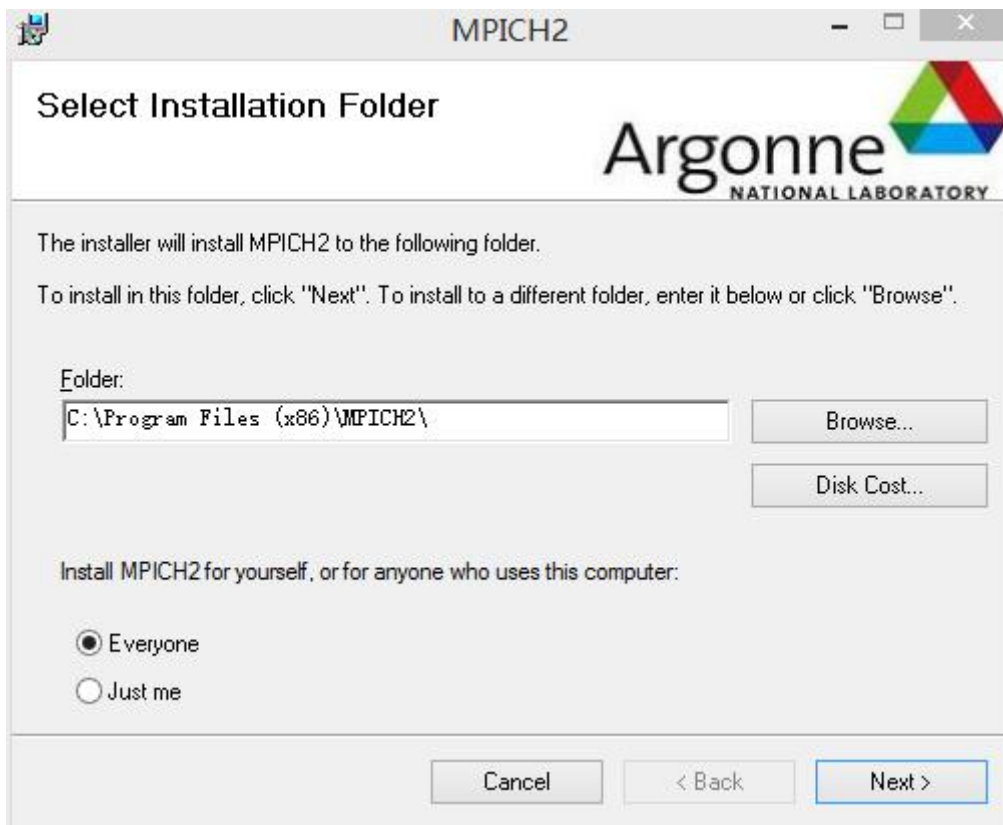
1.2 配置 MPICH

本节将以 MPICH 为例进行配置。（网站：<http://www.mpich.org/>）

1.2.1 MPICH 安装

在 MPICH 网站下可以下载 MPICH2-1.4.1 安装包，区分 64 位与 32 位。双击安装，建议采用默认配置其中需要注意：

1. 设定 MPICH2 所有用户可用



2. 按装时需要赋予管理员权限时点击“确定”
3. SMPD 需要网络访问权限，设置允许
4. 将 `C:\Program Files\MPICH2\bin` 加入 PATH 环境变量中(与 java 类似)

设置完成之后，在 CMD 中执行指令 `where mpiexec` 可以打印出安装路径：

```
C:\Users\neoJames>where mpiexec
C:\Program Files\MPICH2\bin\mpiexec.exe
```

1.2.2 Visual Studio 的编译设置

按正常步骤新建项目，之后进入项目属性，在 VC++ 目录中添加：

包含目录： `C:\Program Files\MPICH2\include`；

库目录： `C:\Program Files\MPICH2\lib`

常规	
可执行文件目录	\$(VCInstallDir)bin;\$(WindowsSDK_ExecutablePath_x86);\$(VSInstallDir)C
包含目录	C:\Program Files (x86)\MPICH2\include;\$(IncludePath)
引用目录	\$(VCInstallDir)atlmfc\lib;\$(VCInstallDir)lib
库目录	C:\Program Files (x86)\MPICH2\lib;\$(LibraryPath) ▼
Windows 运行库目录	\$(WindowsSDK_MetadataPath)
源目录	\$(VCInstallDir)atlmfc\src\mfc;\$(VCInstallDir)atlmfc\src\mfc\m;\$(VCInsta
排除目录	\$(VCInstallDir)include;\$(VCInstallDir)atlmfc\include;\$(WindowsSDK_Incl

在链接器>输入的附加依赖项中添加 mpi.lib

附加依赖项	mpi.lib;%(AdditionalDependencies) ▼
忽略所有默认库	
忽略特定默认库	
模块定义文件	
将模块添加到程序集	
嵌入托管资源文件	
强制符号引用	
延迟加载的 DLL	
程序集链接资源	

1.2.3 使用 MPICH 单机运行 MPI 程序

按照 1.2 的步骤配置完成之后，在 Visual Studio 内可以进行 MPI 编程。

在编译完成程序后，即可运行，在 windows 下运行 mpi 程序需要做如下准备：

1. 注册：

在 wmpi 中注册一个可以登录本 windows 的账号



2. 开启 smpd 服务。

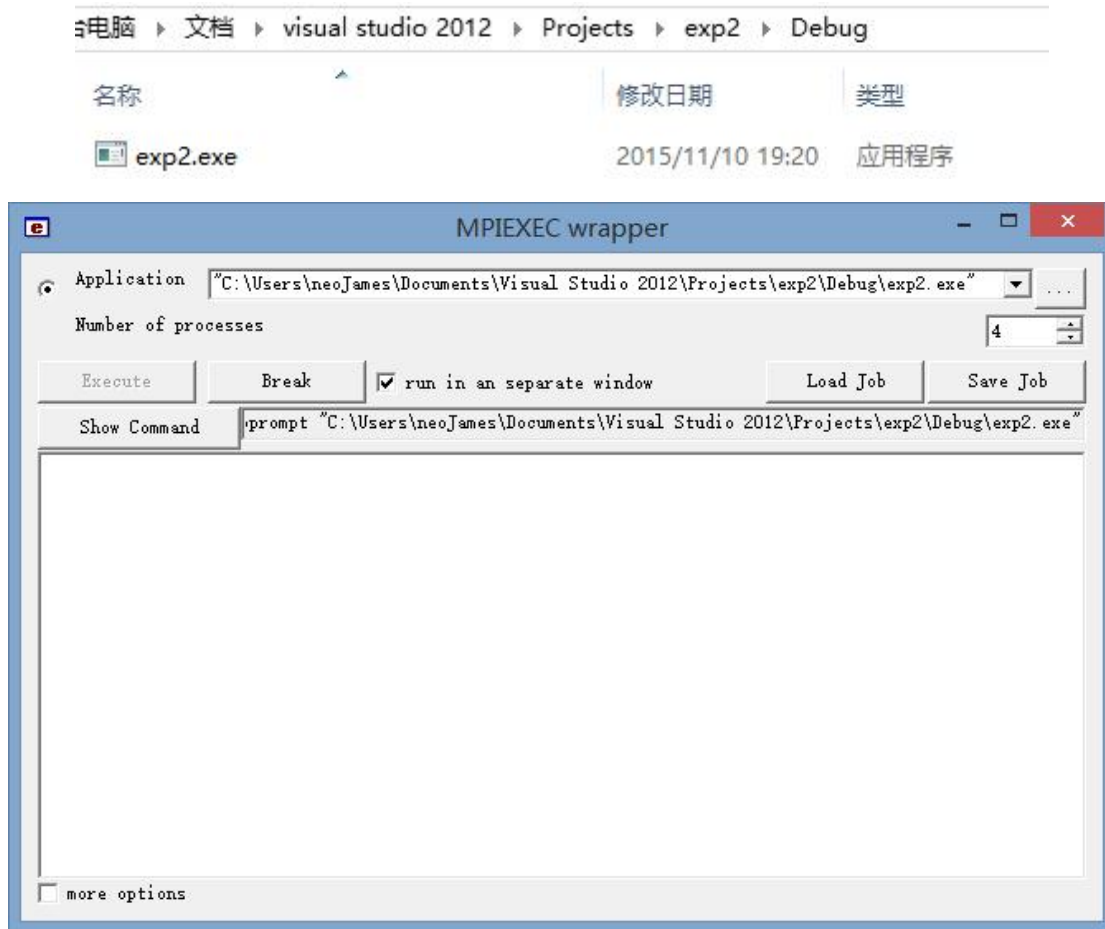
使用管理员身份运行 CMD，运行 `smpd -install -phrase behappy`（behappy 为默认密码）。

3. 开启 wmpiexec 程序，找到编译完成的 visual studio 程序，运行。

以如下程序为例：

```
9 int main(int argc, char *argv[])
10 {
11     int i;
12     int p;
13     int my_rank;
14     MPI_Init(&argc, &argv);          /* 启动MPI计算 */
15     MPI_Comm_size(MPI_COMM_WORLD, &p); /* 确定处理器个数 */
16     MPI_Comm_rank(MPI_COMM_WORLD, &my_rank); /* 确定各自的处理器标识符 */
17
18     if(my_rank==0)
19         printf("pid %d is the master\n", my_rank);
20
21     printf("total processor count is %d, while my pid is %d, \n", p, my_rank);
22
23     MPI_Finalize();                  /* 结束MPI计算 */
24
25     return 0;
26 }
27
```

其存储位置可以通过右键项目->在文件资源管理器中打开文件夹。之后在 Debug 目录下(或 Release 目录下)可以找到可执行文件。



- 此外，也可以在 CMD 中使用 `mpiexec` 执行程序（正确配置环境变量前提下）

第 2 章 Linux 下配置 MPI（以 MPICH 为例）

2.1 MPI 安装

- 下载安装包(指令 `wget <url>`):

```
[zlq@mu01 src]$ wget http://www.mpich.org/static/downloads/3.1.4/mpich-3.1.4.tar.gz
--2015-11-10 19:48:48-- http://www.mpich.org/static/downloads/3.1.4/mpich-3.1.4.tar.gz
Resolving www.mpich.org... 140.221.6.71
Connecting to www.mpich.org|140.221.6.71|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11325036 (11M) [application/x-gzip]
Saving to: "mpich-3.1.4.tar.gz"

12% [==>] 1,421,666 89.7K/s eta 1m 55s
```

- 解压缩

之后 `tar -xzvf mpich***`解压文件到当前目录，之后 `cd` 进解压目录内


```
[zfq@mu01 src]$ cd mpich-3.1.4
[zfq@mu01 mpich-3.1.4]$ ls
aclocal.m4  configure.ac  maint          mpich-doxygen.in  src
autogen.sh  contrib      Makefile.am   mpi.def           subsys_include.m4
CHANGES    COPYRIGHT    Makefile.in   README            test
confdb      doc          man           README.envvar     www
configure  examples    mpich.def     RELEASE_NOTES
[zfq@mu01 mpich-3.1.4]$
```

3. 安装配置

配置默认安装路径（指令 `configure --prefix=xxxx`）：

```
[zfq@mu01 mpich-3.1.4]$ ./configure --prefix=$HOME/usr/mpich
```

之后 `make all`

然后 `make install`

之后 `mpi` 就安装在指定路径（`$HOME/usr/mpich` 路径下）

```
[zfq@mu01 mpich-3.1.4]$ cd ~/usr/mpich/
[zfq@mu01 mpich]$ ls
bin  include  lib  share
[zfq@mu01 mpich]$
```

4. 配置环境变量

环境变量配置方式有很多种，这里用配置 `.bash_profile` 的方式配置：

使用编辑器编辑 `$HOME/.bash_profile`，在其中加入语句：

```
PATH=$HOME/usr/mpich/bin:$PATH
```

```
export PATH
```

```
[zfq@mu01 mpich]$ vim ~/.bash_profile
[zfq@mu01 mpich]$ which mpiexec
/usr/bin/mpiexec
[zfq@mu01 mpich]$ source ~/.bash_profile
[zfq@mu01 mpich]$ which mpiexec
~/usr/mpich/bin/mpiexec
[zfq@mu01 mpich]$
```

```
PATH=$HOME/usr/mpich/bin:$PATH
export PATH
```

配置成功以重新登陆终端后 `which mpiexec` 指令显示自己为自己安装的 `mpi` 目录为准：

```
[zfq@mu01 ~]$ which mpiexec
~/usr/mpich/bin/mpiexec
[zfq@mu01 ~]$
```

2.2 编译与运行（单节点）

直接使用 `mpicxx` 编译，并且使用 `mpiexec` 运行：

```
[zlq@mu01 exp2]$ mpicxx mpi_test.cpp -o mpi_test
[zlq@mu01 exp2]$ mpiexec -n 4 mpi_test
total program number is 4
while my program id is 0
total program number is 4
while my program id is 1
total program number is 4
while my program id is 2
total program number is 4
while my program id is 3
[zlq@mu01 exp2]$
```

第3章 ** (选) 多节点运行 (集群环境)

3.1 配置 SSH

3.1.1 配置 ssh 无密码登陆

1. 生成公钥 (ssh-keygen 指令, 填写默认选项):

```
neo@ubuntu-mu01:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/neo/.ssh/id_rsa):
Created directory '/home/neo/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/neo/.ssh/id_rsa.
Your public key has been saved in /home/neo/.ssh/id_rsa.pub.
The key fingerprint is:
93:c9:18:9e:66:37:e3:dc:2a:05:0d:5a:5c:bb neo@ubuntu-mu01
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      .  .  .      |
|      +   .       |
|     o.o.        |
|    ...=.+.      |
|     *.E         |
|    o +.=        |
|     oo..        |
|    . o ..       |
|     ..o.        |
+-----+

```

2. 将公钥传输到指定机器上 (ssh-copy-id -i ~/.ssh/id_rsa.pub <remotehost>)

```
neo@ubuntu-mu01:~$ ssh-copy-id -i ~/.ssh/id_rsa.pub 192.168.100.101
The authenticity of host '192.168.100.101 (192.168.100.101)' can't be established.
ECDSA key fingerprint is eb:e9:4e:b7:c3:d6:98:8c:67:cc:30:bb:83:96:a3:c1.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.100.101' (ECDSA) to the list of known hosts.
neo@192.168.100.101's password:
Now try logging into the machine, with "ssh '192.168.100.101'", and check in:

  ~/.ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.
```

3. 成功标志为可以无密码登陆:

```
neo@ubuntu-mu01:~$ ssh 192.168.100.101
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.13.0-32-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Thu Nov 12 22:06:33 CST 2015

System load:  0.0                Processes:            72
Usage of /:   13.7% of 6.75GB    Users logged in:     0
Memory usage: 7%                IP address for eth0: 10.0.2.15
Swap usage:   0%                IP address for eth1: 192.168.100.101

Graph this data and manage this system at:
  https://landscape.canonical.com/

New release '14.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2017.

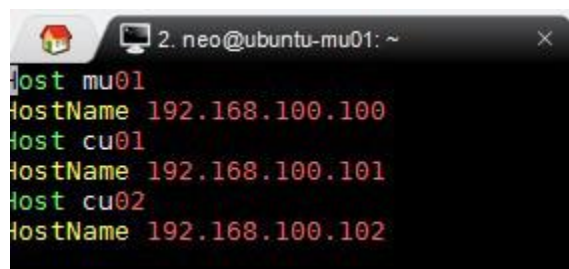
Last login: Thu Nov 12 14:09:15 2015
neo@ubuntu-cu01:~$
```

本步骤需要在所有节点重复。

SSH 协议采用非对称加密，具体技术细节可以在网络安全相关课程学到。

3.1.2 配置 ssh 昵称

编辑 ~/.ssh/config 文件，添加如下字段保存，之后可以使用昵称连接各个计算节点。



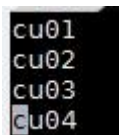
```
Host mu01
HostName 192.168.100.100
Host cu01
HostName 192.168.100.101
Host cu02
HostName 192.168.100.102
```

然后使用 scp 指令进行复制。

3.2 执行程序

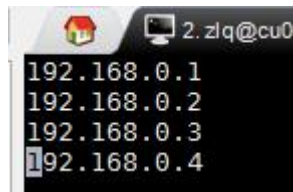
需要多节点执行首先需要保证每个节点都安装了相同版本的 MPI。此外，需要编辑一个 host_file，之后将 hostfile 作为参数传入。

例如编辑 host_file 内容如下（cu01 是 192.168.0.1 的别名）:



```
cu01
cu02
cu03
cu04
```

Or



执行指令：mpirun -n 16 -f host_file mpi_test 可得如下运行结果（各个节点上都有程序运行）：

```
top - 20:43:41 up 75 days, 9:00, 4 users, load average: 2.76, 0.
Tasks: 1043 total, 5 running, 1037 sleeping, 0 stopped, 1 zom
Cpu(s): 12.7%us, 0.1%sy, 0.0%ni, 87.2%id, 0.0%wa, 0.0%hi, 0.0%
Mem: 65906780k total, 62392516k used, 3514264k free, 168688k bu
Swap: 65535996k total, 25236k used, 65510760k free, 56351996k ca

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
26593 zlq 20 0 45124 2620 2212 R 100.0 0.0 1:12.15 mpi_te
20590 zlq 20 0 45124 2620 2216 R 99.9 0.0 1:12.15 mpi_tes
20591 zlq 20 0 45124 2616 2208 R 99.9 0.0 1:12.14 mpi_tes
20592 zlq 20 0 45124 2620 2212 R 99.9 0.0 1:12.14 mpi_tes
19664 cyh 20 0 481m 33m 22m S 7.0 0.1 42:12.46 krfb
443 root 39 19 0 0 0 S 1.0 0.0 1021:00 kipmi0
15431 hongrq 20 0 17.7g 585m 14m S 0.7 0.9 52:11.43 java

top - 20:43:35 up 75 days, 8:56, 3 users, load average: 2.75, 0.
Tasks: 819 total, 6 running, 813 sleeping, 0 stopped, 0 zombi
Cpu(s): 12.5%us, 0.1%sy, 0.0%ni, 87.4%id, 0.0%wa, 0.0%hi, 0.0%
Mem: 65906780k total, 50929356k used, 14977424k free, 108756k bu
Swap: 65535996k total, 9660k used, 65526336k free, 48728424k ca

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
18694 zlq 20 0 45124 2624 2216 R 100.0 0.0 1:12.13 mpi_te
18695 zlq 20 0 45124 2616 2208 R 100.0 0.0 1:12.14 mpi_te
18697 zlq 20 0 45124 2620 2212 R 100.0 0.0 1:12.14 mpi_te
18696 zlq 20 0 45124 2620 2212 R 99.8 0.0 1:12.13 mpi_tes
443 root 39 19 0 0 0 S 1.0 0.0 1046:27 kipmi0
11196 hadoop 20 0 1675m 255m 13m S 1.0 0.4 38:12.46 java
27119 hadoop 20 0 5782m 249m 17m S 0.7 0.4 15:37.34 java

top - 20:43:44 up 75 days, 9:02, 35 users, load average: 4.79, 2.6
Tasks: 1559 total, 6 running, 1548 sleeping, 1 stopped, 4 zom i
Cpu(s): 14.7%us, 2.5%sy, 0.0%ni, 81.9%id, 0.8%wa, 0.0%hi, 0.1% i
Mem: 65906780k total, 29423832k used, 36482948k free, 60648k bu f
Swap: 65535996k total, 3431716k used, 62104280k free, 8231464k ca h

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
2857 root 20 0 4168 136 100 R 100.0 0.0 1492:15 cache.f
26809 zlq 20 0 45124 2624 2216 R 100.0 0.0 1:12.25 mpi_tes
26810 zlq 20 0 45124 2616 2208 R 100.0 0.0 1:12.25 mpi_tes
26812 zlq 20 0 45124 2620 2212 R 100.0 0.0 1:12.25 mpi_tes
26811 zlq 20 0 45124 2624 2212 R 99.8 0.0 1:12.24 mpi_tess
25939 cvstaff 20 0 130m 2772 1760 D 44.8 0.0 5600:41 gvfsd-me
4179 root 0 -20 24612 1428 960 S 1.3 0.0 461:08.41 lim
```

第 4 章 *(选)虚拟机实验（以 Linux 为例）

4.1 环境配置

本配置：

虚拟机使用 VirtualBox-5.0.8

系统使用 Ubuntu-12.04.5-server

MPI 使用 MPICH-3.1.4

SSH 工具使用 MobaXterm

其它任何版本、任何替代配置皆可进行实验，配置有区别本教程不赘述。

具体下载信息如下：

项目	版本	网址
VirtualBox	5.0.8	https://www.virtualbox.org/wiki/Downloads
Ubuntu	12.04.5	http://www.ubuntu.com/download/alternative-downloads
MPICH	3.1.4	https://www.mpich.org/downloads/

4.2 新建虚拟机

安装过程最好联网，可以并行安装多台虚拟机


1. 点击左上角新建，之后按照个人喜好输入一个名字，

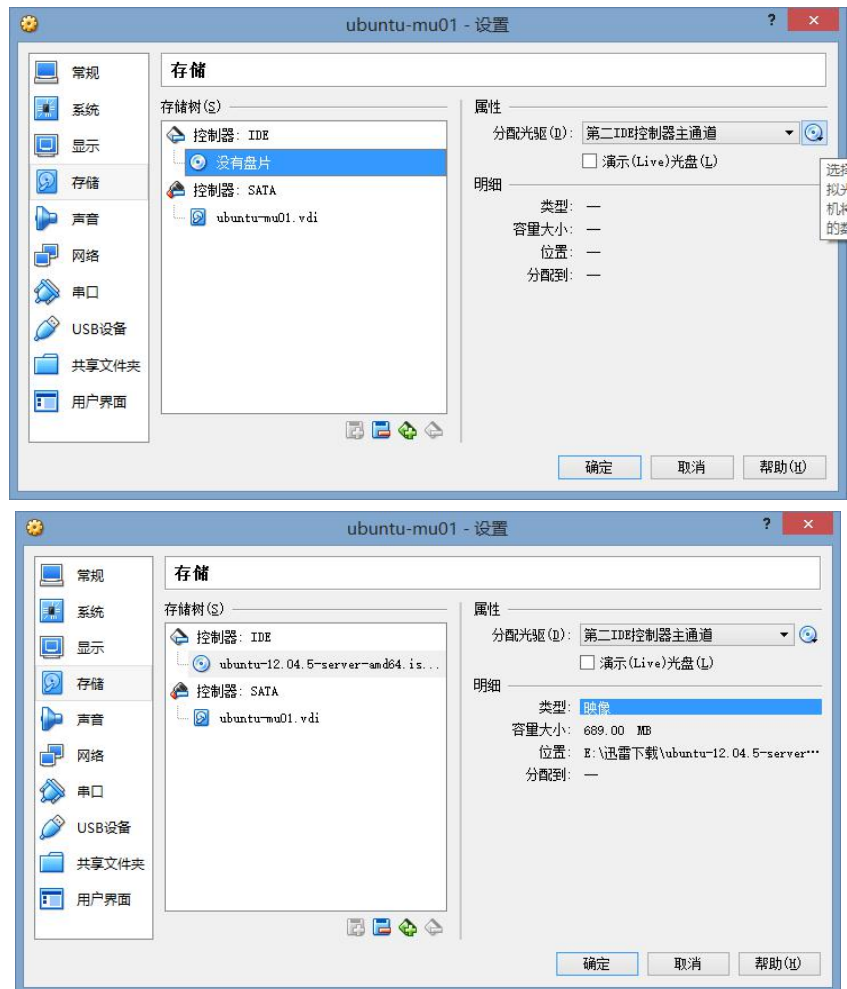


2. 内存建议“768MB”，硬盘大小建议“8GB”，硬盘文件类型建议“VDI”，建议“动态分配”空间（默认设置）



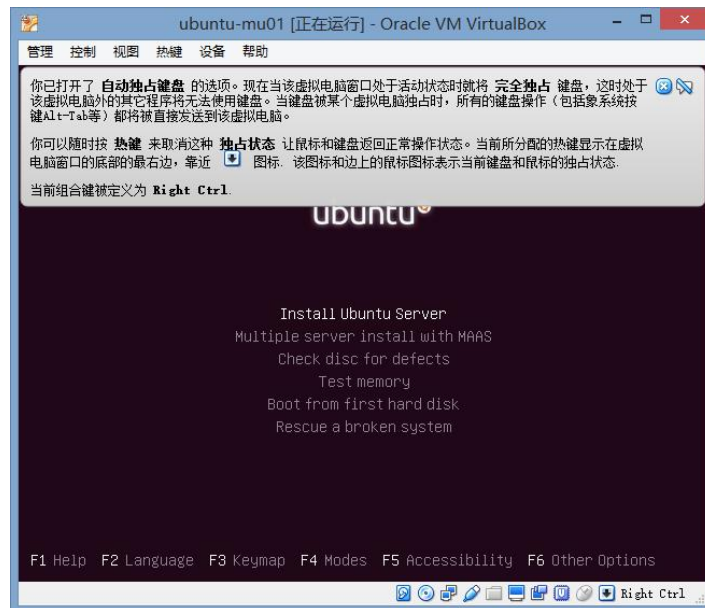
3. 完成之后左边可见新出现一个虚拟机，选择点击“设置”

a) 左栏存储，点击“没有盘片”点击右侧“第二 IDE 控制器主通道”旁边的光盘，选择下载了的镜像

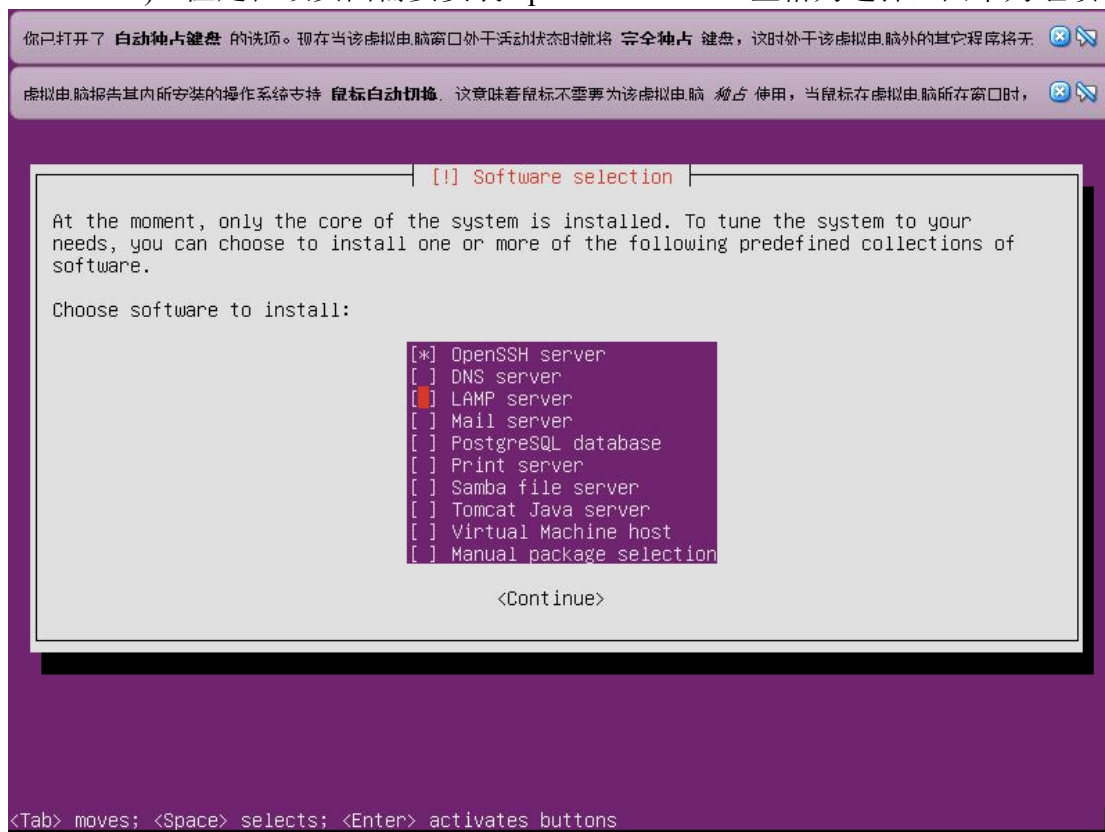


4. 完成之后点击启动，进行系统安装

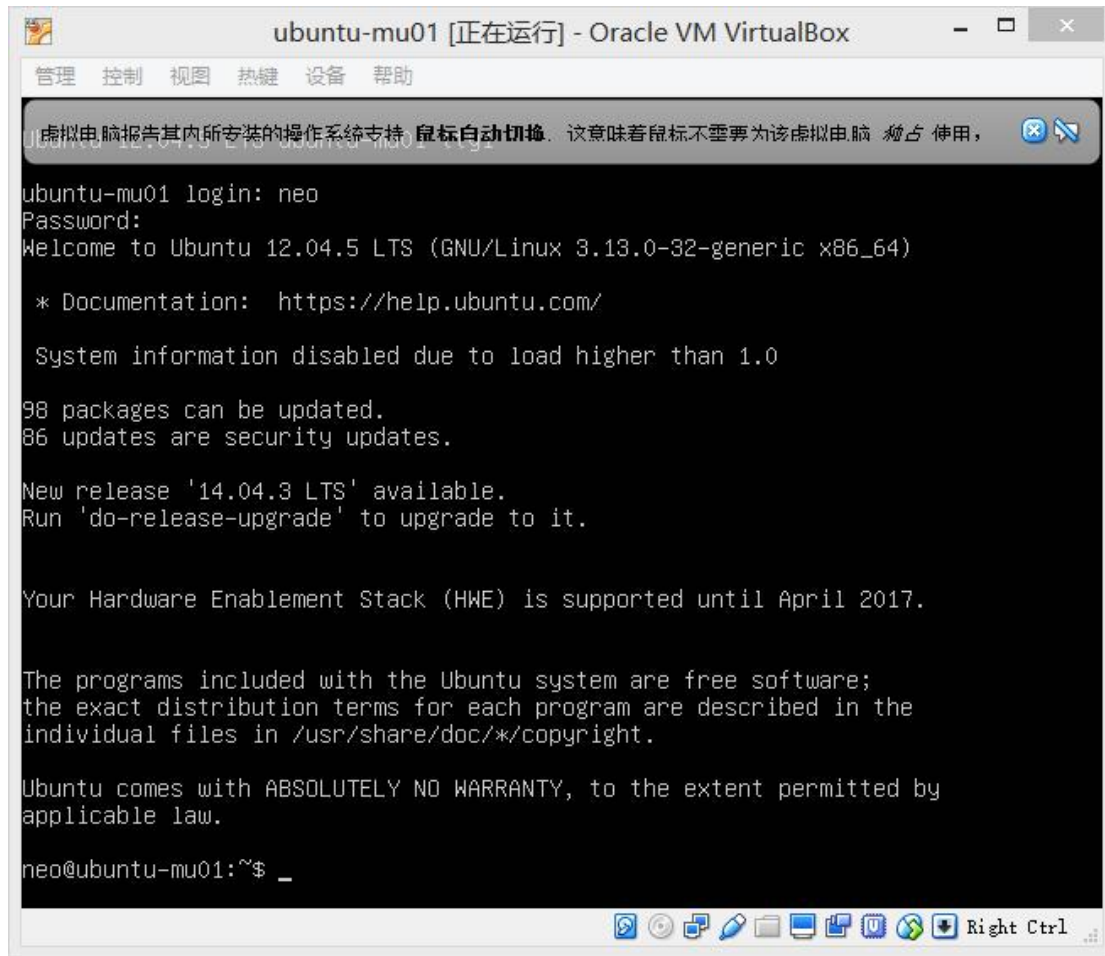




- 除了 hostname 建议和虚拟机名字一样之外，剩下配置与普通安装系统过程无异，此处不赘述
- 但是在该页面需要安装 openSSH server（空格为选择，回车为继续）



安装成功标志：



5. 完安装后，可以再按照常用软件（ssh、vim）：

- a) 输入 `sudo apt-get update`
- b) 输入 `sudo apt-get install vim`

6. 完成安装后，可以按照第二章的教程将 MPI 配置进机器(建议)

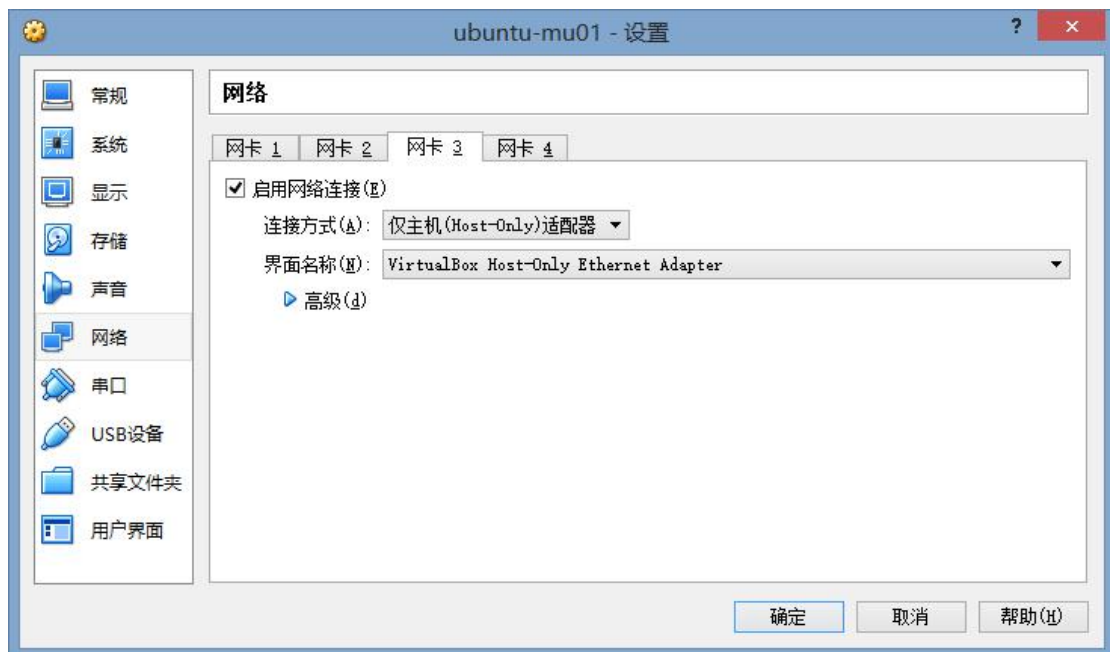
4.3 配置集群网络

4.3.1 已安装多个虚拟机情况：

1. 重新进入安装好的机器的设置中（关机状态）
2. 选择网络，“网卡 1”选择“网络地址转换（NAT）”（保持默认设置），网卡 2>“启用网络”>连接方式“内部网络”>界面名称(随便定一个)

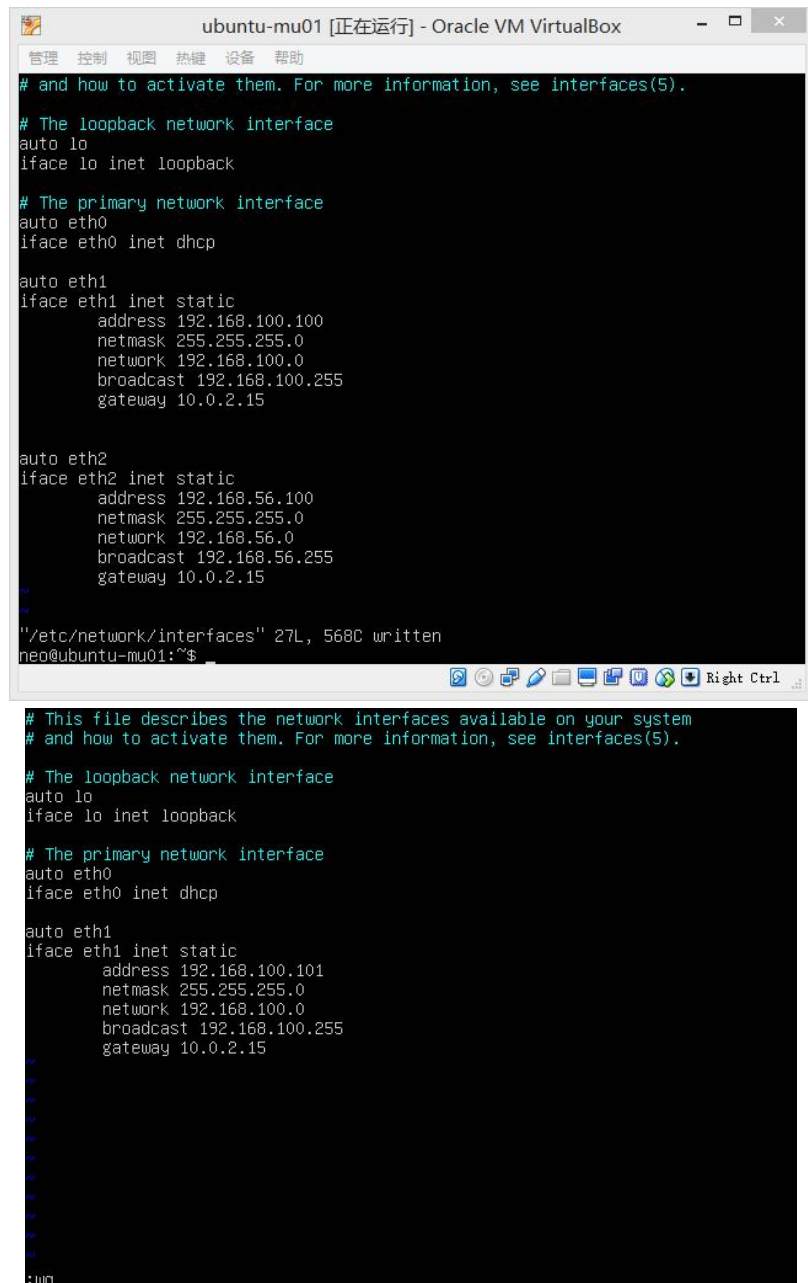


其中，对于主节点，设置网卡 3：



3. 开机，编辑/etc/network/interfaces 设置各个节点 IP 地址，主节点与普通节点设置方式分别展示如下（主节点为 mu01，普通节点为 cu**）

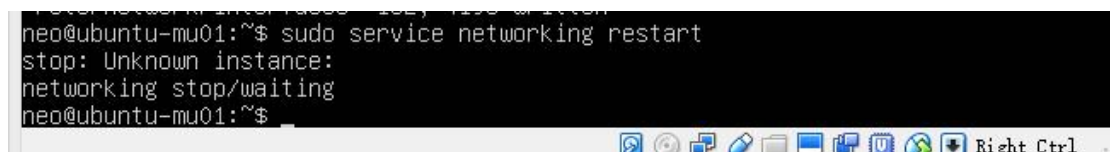
```
neo@ubuntu-mu01:~$ sudo vim /etc/network/interfaces
```



```
ubuntu-mu01 [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0 inet dhcp
auto eth1
iface eth1 inet static
    address 192.168.100.100
    netmask 255.255.255.0
    network 192.168.100.0
    broadcast 192.168.100.255
    gateway 10.0.2.15
auto eth2
iface eth2 inet static
    address 192.168.56.100
    netmask 255.255.255.0
    network 192.168.56.0
    broadcast 192.168.56.255
    gateway 10.0.2.15
"/etc/network/interfaces" 27L, 568C written
neo@ubuntu-mu01:~$

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0 inet dhcp
auto eth1
iface eth1 inet static
    address 192.168.100.101
    netmask 255.255.255.0
    network 192.168.100.0
    broadcast 192.168.100.255
    gateway 10.0.2.15
```

4. 重启 network 服务（普通 linux 使用 `service networking restart`，ubuntu 该命令并不能生效，所以可以 `reboot` 使修改生效）



```
neo@ubuntu-mu01:~$ sudo service networking restart
stop: Unknown instance:
networking stop/waiting
neo@ubuntu-mu01:~$
```

5. 各个节点内网络配置成功标识为相互可以 ping 通

```
neo@ubuntu-cu01:~$ ping 192.168.100.102
PING 192.168.100.102 (192.168.100.102) 56(84) bytes of data.
64 bytes from 192.168.100.102: icmp_req=1 ttl=64 time=0.266 ms
64 bytes from 192.168.100.102: icmp_req=2 ttl=64 time=0.260 ms
^C
--- 192.168.100.102 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.260/0.263/0.266/0.003 ms
neo@ubuntu-cu01:~$ ping 192.168.100.100
PING 192.168.100.100 (192.168.100.100) 56(84) bytes of data.
64 bytes from 192.168.100.100: icmp_req=1 ttl=64 time=0.256 ms
64 bytes from 192.168.100.100: icmp_req=2 ttl=64 time=0.248 ms
64 bytes from 192.168.100.100: icmp_req=3 ttl=64 time=0.259 ms
^C
--- 192.168.100.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.248/0.254/0.259/0.013 ms
neo@ubuntu-cu01:~$
```

6. 主节点网络配置成功标志除了内网可以 ping 通之外，还需要保证外网可以通过 ssh 工具访问。

```
[neoJames.school_pc] > ssh neo@192.168.56.100
Warning: Permanently added '192.168.56.100' (RSA) to the list of known hosts.
neo@192.168.56.100's password:
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.13.0-32-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Thu Nov 12 14:20:06 CST 2015

System load:  0.0               Users logged in:  1
Usage of /:   13.7% of 6.75GB   IP address for eth0: 10.0.2.15
Memory usage: 9%               IP address for eth1: 192.168.100.100
Swap usage:   0%               IP address for eth2: 192.168.56.100
Processes:   78

Graph this data and manage this system at:
  https://landscape.canonical.com/

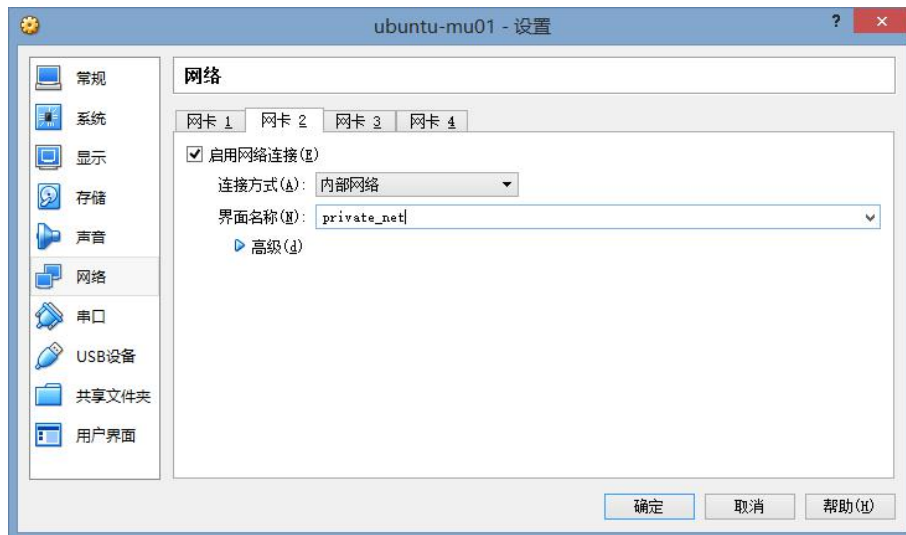
New release '14.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2017.

Last login: Thu Nov 12 14:08:47 2015
/usr/bin/xauth:  file /home/neo/.Xauthority does not exist
neo@ubuntu-mu01:~$
```

4.3.2 安装单个虚拟机需要复制情况:

1. 重新进入安装好的机器的设置中（关机状态）（所有虚拟机全都
 - a) 选择网络，“网卡 1”选择“网络地址转换（NAT）”（保持默认设置），网卡 2>“启用网络”>连接方式“内部网络”>界面名称(随便定一个)



2. 重启系统，使用 ifconfig 查看网络状态：

```
ifneo@ubuntu-mu01:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:30:79:36
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe30:7936/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:22 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2186 (2.1 KB)  TX bytes:2336 (2.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

3. 以管理员权限编辑/etc/network/interfaces 文件：

```
neo@ubuntu-mu01:~$ sudo vim /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
    address 192.168.100.101
    netmask 255.255.255.0
    network 192.168.100.0
    broadcast 192.168.100.255
    gateway 10.0.2.15
```

4. 重启 network 服务（普通 linux 使用 service networking restart，ubuntu 该命令并不能生效，所以可以 reboot 使修改生效）

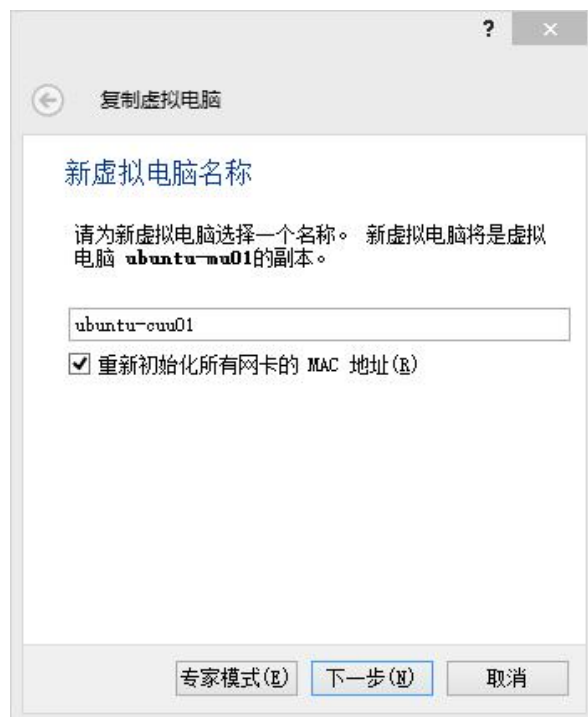
5. 此时查看 ifconfig 可以看到已修改网络:

```
neo@ubuntu-mu01:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:30:79:36
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe30:7936/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:28 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3536 (3.5 KB)  TX bytes:2776 (2.7 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:f9:39:ff
          inet addr:192.168.100.101  Bcast:192.168.100.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe39:39ff/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:648 (648.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

6. 关机, 复制虚拟机 (右键->复制), 按照个人喜好命名 (注意一个机器为主节点)



7. 主节点添加一张网卡设置如下:



8. 开启所有虚拟机，在计算节点，重命名 host-name（建议和虚拟机名字一致）

```
neo@ubuntu-mu01:~$ sudo vim /etc/hostname
```

9. 修改 IP 地址，保证主节点为 192.168.100.100，其余节点不重复即可

```
# and how to activate them. For more information, see interfaces(5).

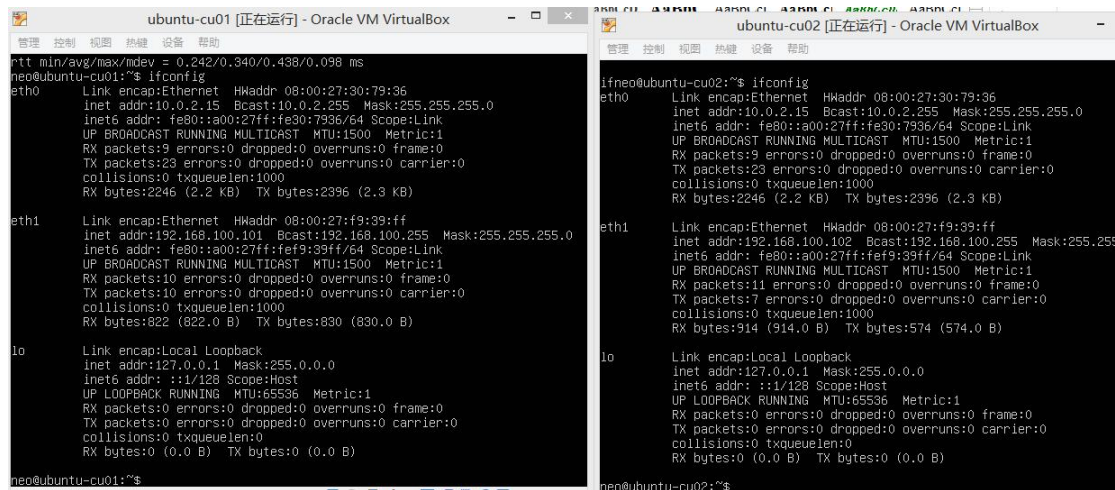
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

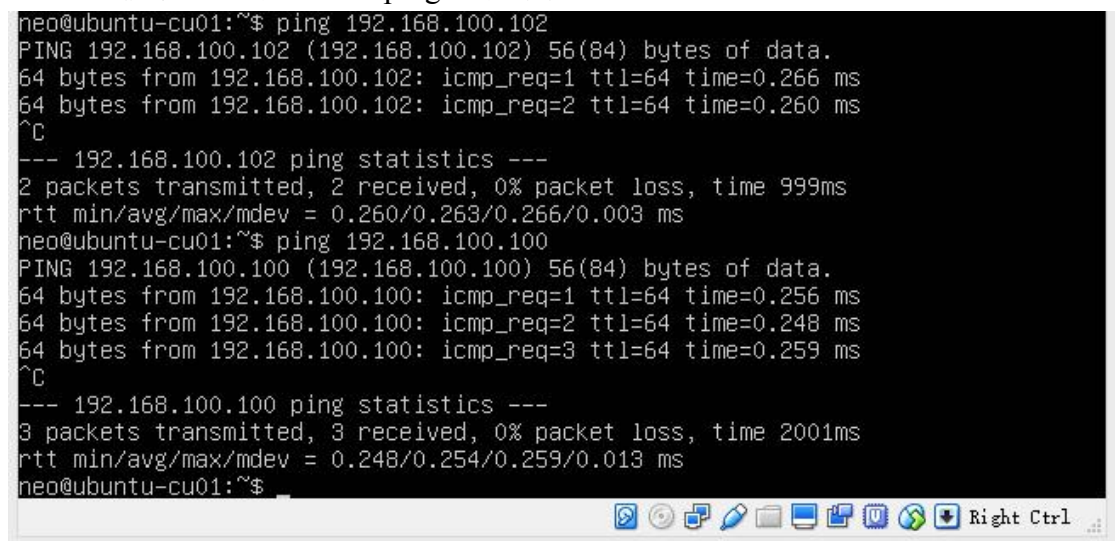
auto eth1
iface eth1 inet static
    address 192.168.100.100
    netmask 255.255.255.0
    network 192.168.100.0
    broadcast 192.168.100.255
    gateway 10.0.2.15

auto eth2
iface eth2 inet static
    address 192.168.56.100
    netmask 255.255.255.0
    network 192.168.56.0
    broadcast 192.168.56.255
    gateway 10.0.2.15

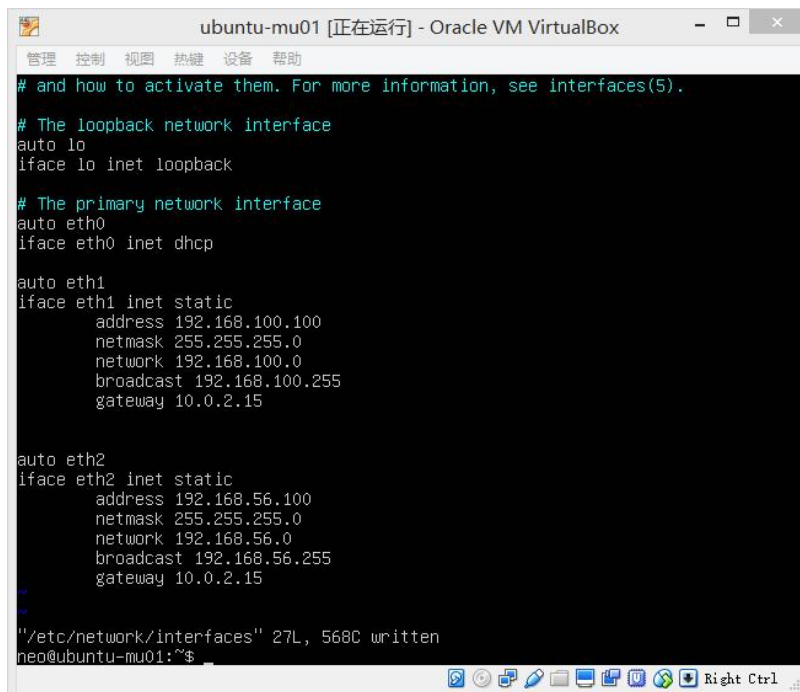
"/etc/network/interfaces" 27L, 568C written
neo@ubuntu-mu01:~$
```



10. 修改的结果以可以 ping 通各个内网地址为准:



11. 主节点设置网卡三的配置方式如下:



12. 配置成功后可以通过 ssh 工具连接主节点：

```
[neoJames.school_pc] > ssh neo@192.168.56.100
Warning: Permanently added '192.168.56.100' (RSA) to the list of known hosts.
neo@192.168.56.100's password:
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.13.0-32-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Thu Nov 12 14:20:06 CST 2015

System load:  0.0           Users logged in:  1
Usage of /:   13.7% of 6.75GB IP address for eth0: 10.0.2.15
Memory usage: 9%           IP address for eth1: 192.168.100.100
Swap usage:   0%           IP address for eth2: 192.168.56.100
Processes:    78

Graph this data and manage this system at:
https://landscape.canonical.com/

New release '14.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

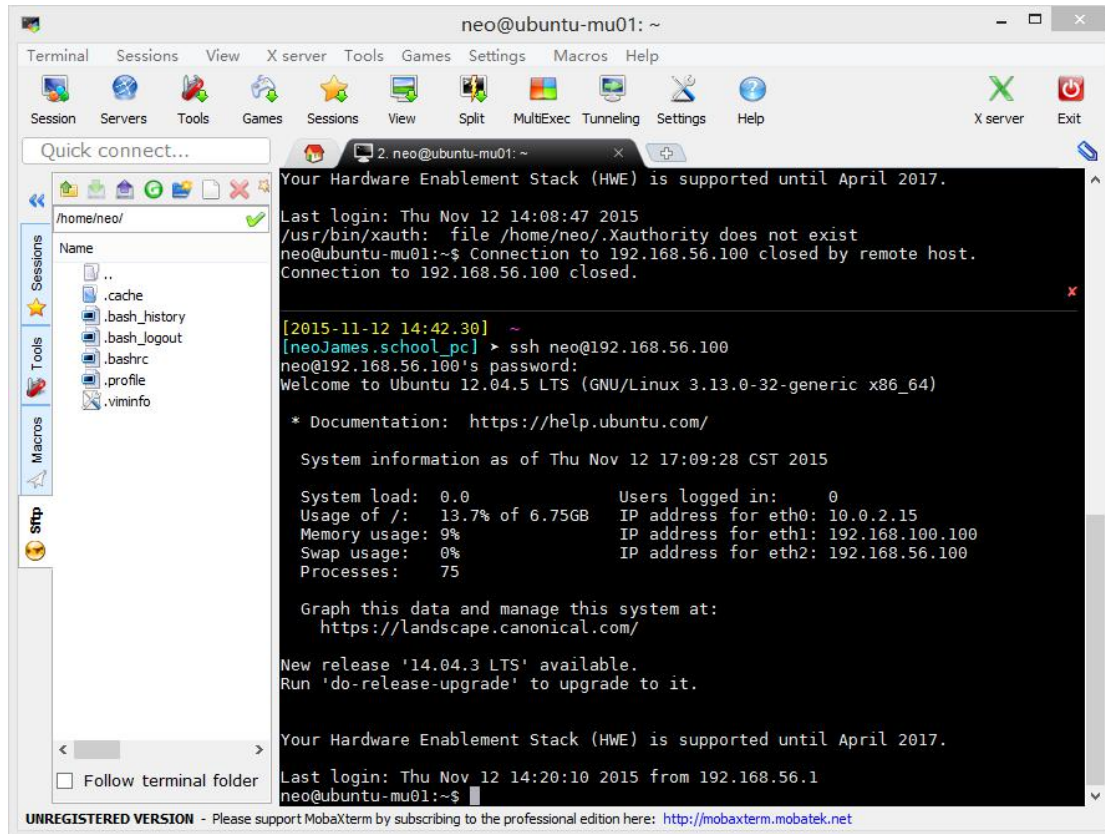
Your Hardware Enablement Stack (HWE) is supported until April 2017.

Last login: Thu Nov 12 14:08:47 2015
/usr/bin/xauth: file /home/neo/.Xauthority does not exist
neo@ubuntu-mu01:~$
```

4.4 配置 SSH（与 3.1 重复）

在网络配置完成之后，不必开启虚拟机界面，直接使用无界面开机，然后通过 ssh 工具登陆：





4.4.1 配置 ssh 无密码登陆

4. 生成公钥（ssh-keygen 指令，填写默认选项）：

```

neo@ubuntu-mu01:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/neo/.ssh/id_rsa):
Created directory '/home/neo/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/neo/.ssh/id_rsa.
Your public key has been saved in /home/neo/.ssh/id_rsa.pub.
The key fingerprint is:
93:c9:18:9e:66:37:e3:dc:2b:dc:2a:05:0d:5a:5c:bb neo@ubuntu-mu01
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      .  .  .          |
|      +   .           |
|     o.o.            |
|    ...=.+.         |
|   *.E              |
|  o +.=             |
|   oo..             |
|   . o ..           |
|   ..o.            |
+-----+
  
```

5. 将公钥传输到指定机器上（ssh-copy-id -i ~/.ssh/id_rsa.pub <remotehost>）

```

neo@ubuntu-mu01:~$ ssh-copy-id -i ~/.ssh/id_rsa.pub 192.168.100.101
The authenticity of host '192.168.100.101 (192.168.100.101)' can't be established.
ECDSA key fingerprint is eb:e9:4e:b7:c3:d6:98:8c:67:cc:30:bb:83:96:a3:c1.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.100.101' (ECDSA) to the list of known hosts.
neo@192.168.100.101's password:
Now try logging into the machine, with "ssh '192.168.100.101'", and check in:

  ~/.ssh/authorized_keys
to make sure we haven't added extra keys that you weren't expecting.

```

6. 成功标志为可以无密码登陆:

```

neo@ubuntu-mu01:~$ ssh 192.168.100.101
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.13.0-32-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Thu Nov 12 22:06:33 CST 2015

System load:  0.0               Processes:            72
Usage of /:   13.7% of 6.75GB   Users logged in:     0
Memory usage: 7%               IP address for eth0: 10.0.2.15
Swap usage:   0%               IP address for eth1: 192.168.100.101

Graph this data and manage this system at:
  https://landscape.canonical.com/

New release '14.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2017.

Last login: Thu Nov 12 14:09:15 2015
neo@ubuntu-cu01:~$

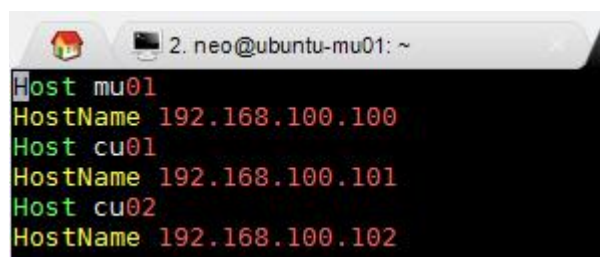
```

本步骤需要在所有节点重复。

SSH 协议采用非对称加密，具体技术细节可以在网络安全相关课程学到。

4.4.2 配置 ssh 昵称

编辑 ~/.ssh/config 文件，添加如下字段保存，之后可以使用昵称连接各个计算节点。



```

Host mu01
HostName 192.168.100.100
Host cu01
HostName 192.168.100.101
Host cu02
HostName 192.168.100.102

```

然后使用 scp 指令进行复制

之后可以按照第 2 章、第 3 章介绍的方式编译和运行 MPI