# CentOS

## 安装

CentOS6.3安装有问题，CentOS6.4没问题。CentOS5.8分为2个DVD，首先在New->Virtual Machine中new CD/DVD(IDE)路径添加DVD1，在安装CentOS提示时，在VMWare的CentOS标签右击setting，更换路径为DVD2，并勾选connected。选择现在定制软件包，选上GNOME和开发工具。

选择系统的语言，英文最稳定。

选择安装哪种键盘，一般选择U.S.English(即美式的)。

Boot分区100mb，swap分区一般选择内存的1.5～2倍。

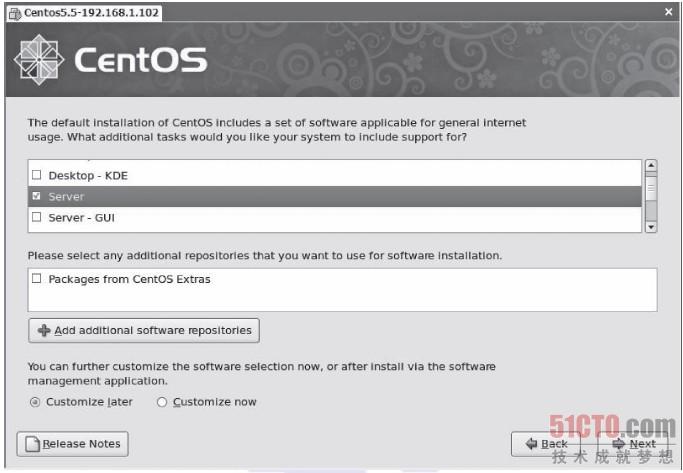
配置系统的网卡时，记得选上“Active on Boot”。如果不选择，会直接导致系统在启动时网卡分配不了IP。配置IP地址、子网掩码、网关及主从DNS服务器。

[](http://images.51cto.com/files/uploadimg/20111107/191846770.jpg)

选择系统时区，一般选择“亚洲/上海”。 “System clock uses UTC”。GPS系统中有两种时间，一为UTC(世界协调时间(Universal Time Coordinated))，另一为CST(地方时)。两者的区别在于时区不同，UTC指的是0时区的时间，地方时为本地时间，如北京为早上8点(东八区)，UTC时间就为零点，时间比北京时间晚8个小时

要牢记一个原则，软件包越少的系统越稳定。

安装系统的软件包选择Server，即最小化安装。

[](http://images.51cto.com/files/uploadimg/20111107/192048420.jpg)

系统安装完成后重启。接着就要选择系统的安全及网络配置。选择“Firewall configuration”，关闭iptables和SELinux。

[](http://images.51cto.com/files/uploadimg/20111107/192203701.jpg)

### 安装GNOME/KDE

显示系统已经安装的组件，和可以安装的组件:

#yum grouplist

如果系统安装之初采用最小化安装，没有安装xwindow，那么先安装：

#yum groupinstall "X Window System"

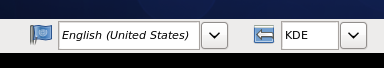
安装GNOME桌面环境

yum groupinstall "GNOME Desktop" "Graphical Administration Tools"

安装KDE桌面环境

yum groupinstall "KDE Desktop"

安装好后登陆界面可选择



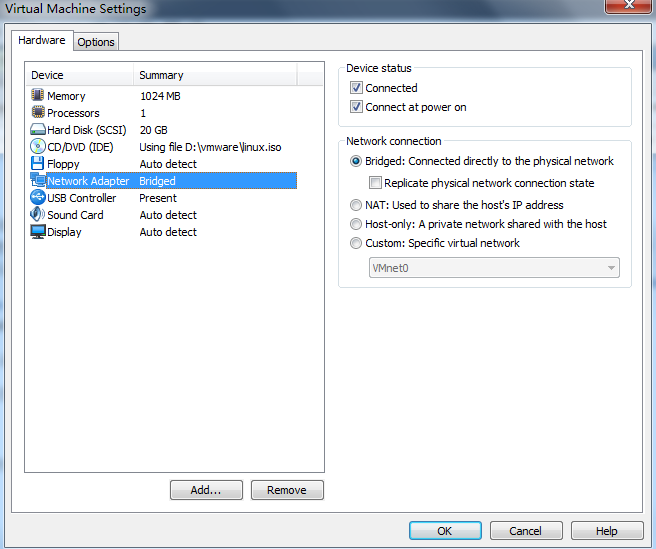
从GNOME切换到KDE桌面

**startkde &**

ln -sf /lib/systemd/system/runlevel5.target /etc/systemd/system/default.target

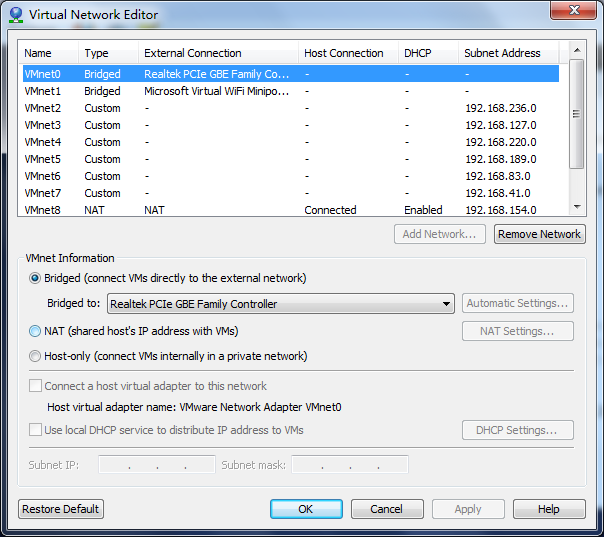
## 桥接

CentOS上右击选setting，在Network connection中选择Bridged



Edit->Virtual Network Editor，选Bridged

不要选Automatic，会自动用无线上网，

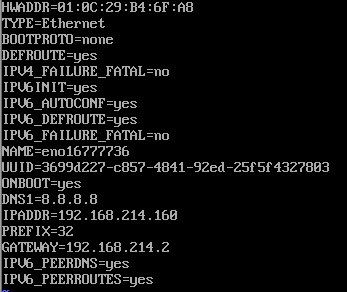


## 设置网络

### CentOS 7

http://www.server-world.info/en/note?os=CentOS\_7&p=initial\_conf&f=3

[root@centos ~]# vi /etc/sysconfig/network-scripts/ifcfg-eno16777736



[root@node0 ~]# ping –c 3 192.168.214.2 //ping通网关

[root@centos ~]# ping –c 5 [www.baidu.com](http://www.baidu.com)

### Ubuntu 14

sudo vi /etc/network/interfaces

auto eth0

iface eth0 inet static

address 192.168.1.3

netmask 255.255.255.0

gateway 192.168.1.1

sudo vim /etc/resolvconf/resolv.conf.d/head

nameserver 1.2.4.8

nameserver 8.8.8.8

sudo resolvconf -u

sudo /etc/init.d/networking restart

**问题1：突然无法登陆putty和samba**

虚拟机Edit->virtual network editor,bridge网卡不选Automatic，选自己的网卡，否则会自动切换到无线网卡等，将本机网络选择固定链接

**问题2：ping: unknown host**

DNS错误

**问题4：Bringing up interface eth0: Error: No suitable device found: no device found for connection 'System eth0'.**

[root@node0 ~]# vi /etc/udev/rules.d/70-persistent-net.rules

SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?\*", ATTR{address}=="08:00:27:25:d3:1c", ATTR{type}=="1", KERNEL=="eth\*", NAME="eth0"

这里的mac物理地址与/etc/sysconfig/network-script/ifcfg-eth0不匹配时，会报该错

**1、修改虚拟机的\*.vmx文件**

这种方法最值得推荐，因为这样就类似于重新"烧录"了VMware虚拟机的"物理网卡ROM"。方法是：

分两种情况：

a：

ethernet0.addressType = "static"

ethernet0.Address = "00:50:56:0A:0B:0C"

"static"说明VM的"物理网卡"的MAC是静态设定的，你可以改成一个以005056开头的另外一个MAC即可。改完启动VM时如果问你SSID的话，选择"Keep Always"。

b：

ethernet0.addressType = "generated"

uuid.location = "56 4d dc f1 ff aa 75 ea-f1 b9 ee 0d 68 9c 65 5c"

uuid.bios = "56 4d ed 23 13 8c 96 91-7c 68 b2 09 8b aa bb cc"

ethernet0.generatedAddress = "00:0c:29:aa:bb:cc"

"generated"说明VM的"物理网卡"的MAC是系统随机动态设定的，你可以通过将uuid.bios后六位及ethernet0.generatedAddress后六位改成你想要改成的以000c29开头的MAC即可。

**2、修改Linux系统里相关 /etc/sysconfig/network-scripts/ifcfg-eth0文件MAC值**

vi /etc/sysconfig/network-scripts/ifcfg-eth0

MACADDR=xx:xx:xx:xx:xx:xx

**3、修改Linux系统里相关rc.local文件MAC值**

可以通过修改启动脚本/etc/rc.d/rc.local：

vi /etc/rc.d/rc.local

ifconfig eth0 down

ifconfig eth0 hw ether 08:00:27:27:19:60

ifconfig eth0 up

**不知道网关**

route -n

**startx vs init 5**

startx：以当前身份进入图形界面

init 5：重启系统，重新输入用户名密码，资源消耗大

# RedHat

**问题1：32位虚拟机安装64位Redhat**

进入BIOS，进入CPU，启动虚拟化

**问题2：yum Unable to read consumer identity**

Redhat没有注册的缘故

**问题3：开机启动失败，/dev/sda3 unexected inconsistency**

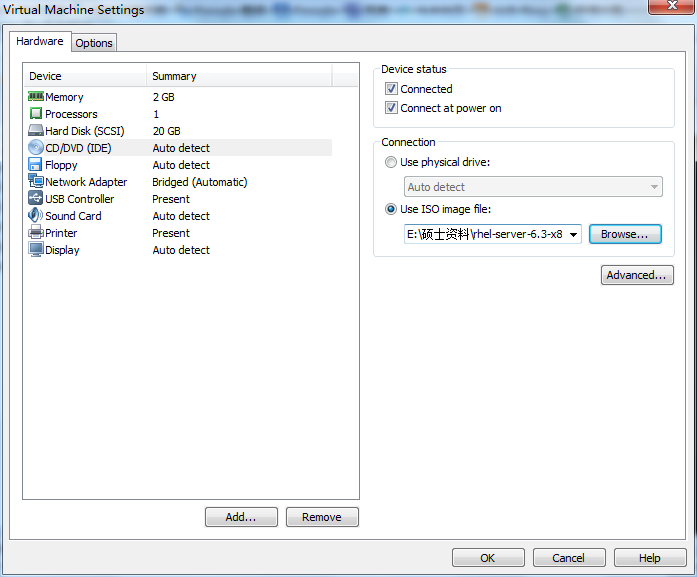
重设系统时间造成的错误

fsck –y /dev/sda3;reboot

# YUM

1. **利用虚拟机挂载光盘**

将iso选入Use ISO image file， 选上Device status Connected。该方法必须进入图形界面，才可yum。桌面出现光盘标志。





1. **利用iso文件**

//如果是DVD光盘(插入服务器后自动映射在/dev/cdrom下), 则先制作ISO文件

[root@ node0 rhel-dvd]# cp /dev/cdrom /media/rhel-dvd/rhel-server-6.3-x86\_64-dvd.iso

[root@ node0 media]# mkdir /media/rhel

[root@ node0 rhel-dvd]# mount -o loop /home/lsfadmin/software/rhel-server-6.3-x86\_64-dvd.iso /media/rhel-dvd

[root@ node0 media]# df -h

Filesystem Size Used Avail Use% Mounted on

/home/lsfadmin/software/rhel-server-6.3-x86\_64-dvd.iso 3.5G 3.5G 0 100% /media/rhel-dvd

1. **更新yum源**

[root@node0~]# cd /media

[root@node0 media]# ln -s /media/RHEL\_6.3\ x86\_64\ Disc\ 1/ rhel-dvd

[root@node0~]# cd /etc/yum.repos.d/

[root@node0 yum.repos.d]# vi rhel-dvd.repo

[rhel-dvd]

name=Red Hat Enterprise Linux $releasever - $basearch - DVD

baseurl=file:///media/rhel-dvd/Server

enabled=1

gpgcheck=1

[root@node1 yum.repos.d]# yum update

# VMwaretools

屏幕下方出行提示安装VMwaretools，单击Install，下载VMwareTools-8.1.4-2227600.tar.gz。

[root@localhost ~] cd /media/VMware\ Tools/

[root@localhost VMware Tools] tar zxvf VMwareTools-8.1.4-2227600.tar.gz -C /home/dwt/software #必须加-C，media目录下无法解压

[root@localhost vmware-tools-distrib]# ./vmware-install.pl 安装VMware Tools

下面的选项就按照默认值。一直回车就可以。屏幕分辨率选1024×768。reboot重启。

# BIOS

虚拟机启动时单击鼠标进入屏幕，然后按F2

或者VM->Power->Power on to BIOS

实体机看蓝屏下面的提示

Redhat启动按F9

# FireFox

## 安装Flash插件

下载install\_flash\_player\_11\_linux.i386.tar.gz

解压$tar zxvf install\_flash\_player\_11\_linux.i386.tar.gz

得到libflashplayer.so和user目录；

$cp libflashplayer.so /usr/lib/mozilla/plugins

如果是64位机器，则拷贝到/usr/lib64/mozilla/plugins目录

$chown root.root /usr/lib/mozilla/plugins/libflashplayer.so

$chmod 755 /usr/lib/mozilla/plugins/libflashplayer.so

重启Firefox

找个flash文件用firefox打开

在firefox中输入about:plugins查看是否有Shockwave Flash

输入about:config，Filter输入dom.ipc.plugins.enabled，其值为true；

## 安装Java插件

下载java，<http://www.java.com/zh_CN/download/linux_manual.jsp?locale=zh_CN>

jre-7u5-linux-i586.rpm或tar.gz两种

[root@centos ~]# rpm –ivh jre-7u5-linux-i586.rpm

inflating: jre-6u25-linux-i586.rpm

Preparing... ########################################### [100%]

1:jre ########################################### [100%]

安装目录为 /usr/java

转至 Firefox 安装目录下的 plugins 子目录

[root@centos ~]# cd /usr/lib/mozilla/plugins

创建指向浏览器插件目录中 libnpjp2.so 文件的符号链接

[root@centos plugins]# ln -s /usr/java/jre1.7.0\_05/lib/i386/libnpjp2.so

[root@centos plugins]# ll

lrwxrwxrwx 1 root root 42 05-23 11:12 libnpjp2.so -> /usr/java/jre1.6.0\_25/lib/i386/libnpjp2.so

启动 Firefox 浏览器，如果 Firefox 浏览器已打开，则重新启动它。

在 Firefox 的地址栏中键入 about:plugins，确认是否已经载入 Java Plugin。您还可以单击“工具”菜单以确认 Java 控制台位于此处。

# NFS

## CentOS

1. 服务器(192.168.0.62)配置

[root@node0 ~]# yum install portmap //必须安装两个软件portmap（注册端口）和nfs

[root@node0 ~]# yum install nfs-utils

[root@node0 ~]# vi /etc/exports //向客户端开放权限

/home 192.168.0.63(rw,no\_root\_squash,sync)

[root@node0 ~]# exportfs //修改exports文件后不用重启nfs

[root@node0 ~]# service portmap start

[root@node0 ~]# service nfs start

[root@node0 ~]# showmount –e 192.168.0.62 //显示共享目录

1. 客户端(192.168.0.63)配置

[root@node1 ~]# service portmap start //服务器端和客户端都必须启动这两服务

[root@node1 ~]# service nfs start

**方法1：临时挂载**

[root@node1 ~]# mount –t nfs 192.168.0.62:/home /home

[root@node1 ~]# df

**方法2：开机挂载，缺点是启动顺序有限制，先启动服务器，再启动客户端**

[root@node1 ~]# vi /etc/fstab

192.168.0.62:/home /home nfs defaults 0 0

[root@node1 ~]# mount -a

**方法3：autofs挂载，自动挂载目录**

**注意：/home会隐藏/home目录下内容，导致dwt无法登陆**

autofs的主要配置文件有两个，分别是/etc下的auto.master和auto.misc。其中，auto.master是起控制作用的，它定义了挂在点和automount动作的文件。

[root@node1 ~]# vi /etc/auto.master

/misc /etc/auto.misc

/home /etc/auto.home //添加该行

[root@node1 ~]# vi /etc/auto.home

dwt -rw,soft,intr 192.168.0.62:/home/dwt

//以下的挂载分别挂载的是nfs,cdrom,floppy,windows共享，本地文分区。

nfs -ro,soft,intr 172.16.0.3:/pub/syd168

cd -fstype=iso9660,iocharset=cp936,ro :/dev/cdrom

fd -fstype=vfat :/dev/fd0

win -fstype=smbfs ://10.8.22.18/syd168

local -fstype=ext3 :/dev/hda1

[root@node1 ~]# service autofs start

**问题1：mount –a时，Server Permission Deny，Server重启NFS。**

**问题2：在node1上以root进入/home/dwt，提示permission deny，是由于/etc/exports以no\_root\_squash挂载，以保留root权限。**

**问题3：mount.nfs: mount point does not exist**

**没有在客户端mkdir目录**

## RedHat

[root@node0 ~]# yum install nfs\* （没有portmap）

[root@node0 ~]# chkconfig nfs on

[root@node0 ~]# chkconfig nfs –list

[root@node0 ~]# vi /etc/exports （内容与CentOS相同）

[root@node0 ~]# service nfs start

[root@node1 ~]# showmount -e 192.168.0.62

**问题1：clnt\_create: RPC: Port mapper failure - Unable to receive: errno 113**

防火墙阻隔，关闭防火墙或添加端口

[root@node0 lsf]# service iptables stop

Node1操作与CentOS相同，**NFS必须保证两台机器上都先mkdir同一目录**

# SSH

Home为共享目录，两台机器以dwt登陆同一目录

[root@node0 ~]# vi /etc/hosts

192.168.0.62 node0 //添加ip与名称对应

192.168.0.63 node1

[dwt@node0 ~]# /etc/init.d/sshd restart //一般默认启动

[dwt@node0 ~]# netstat -tulp | grep sshd

[dwt@node0 ~]# ssh dwt@node1 //登陆，用户名+机器名，从/etc/hosts找出对应ip，密码保存在known\_hosts中

第一次登陆需要保存密匙，输入yes

**无密码ssh登录：**

[dwt@node0 ~]# ssh-keygen -t rsa //产生root的公匙对

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /root/.ssh/id\_rsa.

Your public key has been saved in /root/.ssh/id\_rsa.pub.

The key fingerprint is:

bd:44:42:6d:36:cc:18:83:19:3e:16:8b:42:fc:b5:9a root@node0

[dwt@node0 ~]# ll ~/.ssh

总计 12

-rw------- 1 dwt dwt 1675 08-24 15:17 id\_rsa

-rw-r--r-- 1 dwt dwt 392 08-24 15:17 id\_rsa.pub

-rw-r--r-- 1 dwt dwt 400 08-24 15:04 known\_hosts

[dwt@node0 ~]# scp ~/.ssh/id\_rsa.pub node1:/home/dwt

[dwt@node1 ~]# cat /root/id\_rsa.pub >> /root/.ssh/authorized\_keys

由此，node0可无密码登录node1，node1也需重复以上操作

**问题1：登陆仍需输密码？**

[dwt@node0 .ssh]$ chmod 600 ~/.ssh/authorized\_keys

### Ubuntu 14

sudo apt-get install openssh-server

sudo /etc/init.d/ssh start

sudo update-rc.d ssh defaults

# RSH

[root@node0 ~]# rpm -qa | grep rsh //检测是否已安装rsh和rsh-server

rsh-0.17-60.el6.x86\_64

rsh-server-0.17-60.el6.x86\_64

[root@node0 ~]# yum install rsh\* //安装rsh，rsh-server，xinetd

[root@node0 ~]# service xinetd restart

[root@node0 ~]# netstat -an | grep 514

[root@node0 ~]# echo rsh >> /etc/securetty

[root@node0 ~]# echo rlogin >> /etc/securetty

[root@node0 ~]# echo rexec >> /etc/securetty

[root@node0 ~]# setup

按空格选上rexec,rlogin,rsh

[root@node0 ~]# vi .rhosts

node0

node1

[root@node0 ~]# vi /etc/hosts.equiv

node0

node1

[root@node0 ~]# service xinetd restart

Node1也执行相同操作

rlogin登陆过程中，先检查/etc/hosts.equiv ，看是否存在对应主机和用户，再检查当前用户主目录下的rhosts文件。

因此/etc/hosts.equiv是全局配置，其内容对全部用户都有效，~/.rhosts是局部配置，只对当前用户有效。

普通用户能够无密码的rsh，rlogin不支持root用户的无密码登陆

# MPICH

编译安装：

[root@node0 software]# tar zxvf mpich2-1.4.1p1.tar.gz

[root@node0 software]# yum install gcc\* -y //如果没有安装gcc

[root@node0 software]# cd mpich2-1.4.1p1

[root@node0 mpich2-1.4.1p1]# ./configure --prefix=/home/dwt/software/mpich --with-pm=smpd —with-comm=shared

**注意：pm是mpich的进程管理器，默认使用hydra ，而hydra会出现单机可以正确执行，而多机直接无法通信的问题。**

[root@node0 mpich2-1.4.1p1]# make

[root@node0 mpich2-1.4.1p1]# make install

[dwt@node0 ~]$ vi .bashrc //修改路径

export PATH=/home/dwt/ software/mpich /bin:$PATH

[dwt@node0 ~]$ source .bashrc

[dwt@node0 ~]$ which mpicc

[dwt@node0 ~]$ which mpiexec

[dwt@node0 ~]$ smpd –s //所有节点都要运行smpd –s，第一次运行需输入密码，所有节点密码必须相同

[dwt@node0 ~]$ vi machinefile

node0:3 //mpiexec的配置文件，参与运算的节点：每个节点分配的进程数

node1:2

[dwt@node0 examples]$ mpiexec –machinefile ~/.machinefile –n 5 ./cpi

**注意：节点间必须能互相ssh无密码登录，将home目录挂载到所有节点上，节点上统一用户名为dwt**

**问题1：abort： can’t connect to node1 ，关闭防火墙**

**问题2：Fatel Error MPI\_Send: MPID\_nem\_tcp\_connpoll(1847)**

节点间通信问题，解决方法具体未知，选上所有安装包

/etc/hosts修改为如下形式

192.168.0.62 node0

192.168.0.63 node1

127.0.0.1 localhost.localdomain localhost

::1 localhost6.localdomain6 localhost6

[root@node0 media]# service network restart

**问题3：报错**

op\_connect error: Error = -1

unable to connect mpiexec tree, Error = -1

smpd –s没启动或启动出错

查看是否能无密码互相ssh

**Alternate Process Managers(可选择的PM)**

1. Hydra

利用ssh，rsh，pbs等daemon开始并行作业的进程管理器。

Hydra是MPICH2-1.3默认的PM，mpiexec自动使用Hydra。

Hydra的可执行程序保存在mpich2/bin中，需添加入PATH。

shell$ cat hosts

# This is a sample host file

donner:2 # The first 2 procs are scheduled to run here

foo:3 # The next 3 procs run on this host

shakey:2 # The last 2 procs run on this host

shell$ mpiexec -f hosts -n 4 ./app

Hydra将shell的环境变量传给mpiexec，然后送给各进程。但有些启动器无法自动传播一些环境变量，例如ssh不能传递DISPLAY，SLURM不能传递SLURM等。用户可用-genv强制传递环境变量。

HYDRA\_HOST\_FILE：当mpiexec未提供-f选项时，默认使用的host文件。

For bash:

export HYDRA\_HOST\_FILE=<path\_to\_host\_file>/hosts

For csh/tcsh:

setenv HYDRA\_HOST\_FILE <path\_to\_host\_file>/hosts

HYDRA\_DEBUG：1为使用调试模式，0为不使用；

HYDRA\_ENV：all为传递所有节点环境到各进程，默认不设置，启动节点的环境会传递给可执行程序，只要它的环境变量不变。

HYDRA\_LAUNCH\_EXTRA\_AGES：启动器附加的环境变量值，比如启动器增加-q参数，则HYDRA\_LAUNCH\_EXTRA\_AGES=-q。

MPIEXEC\_TIMEOUT：作业运行执行的最长时间，，时间结束，作业停止。

MPIEXEC\_PORT\_RANGE：如果设置该变量，Hydra将限制端口使用范围，如果未设置，但设置了MPICH\_PORT\_RANGE，则用指定的MPICH\_PORT\_RANGE作为端口，否则它将使用任意端口。端口范围为以冒号分隔的整数对。

HYDRA\_PROXY\_RETRY\_COUNT：代理服务器尝试连接主服务器的次数。

1. Mpd

MPD是传统PM，mpich2-1.4.1p1/src/pm/mpd/README，已弃用。

1. Smpd

Smpd是Windows和Unix交互的PM，mpich2-1.4.1p1/src/pm/smpd/README。

每个节点上都必须运行smpd daemon，在所有参与运算的节点中运行smpd –s。

默认配置文件在~/.smpd。可通过-smpdfile重设smpd文件名，当前用户必须可读写该文件。

如果默认smpd文件不存在，则smpd会要求生成一个并指定密码。

密码用于进入smpd，不要设为Unix密码，所有密码必须相同。

Smpd文件格式为：

phrase=<passphrase>

hosts=<host1 host2 host3 ...>

log=<yes or no>

logfile=<path/filename>

关闭smpd：smpd –shutdown

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

usage: smpd [-port <portnum>] [-phrase <passphrase>] \

[-debug] [-noprompt] [-restart] [-shutdown] \

[-shutdown <hostname>] [-smpdfile <filename>]

-port tells the smpd which port to listen on. The default is 8675

-phrase is used to specify a passphrase used to authenticate the connection with

an smpd

-debug yields lots of output from the smpd.

-noprompt prevents smpd from prompting for information if it is needed. Instead

it returns an error.

-restart restarts the local smpd

-restart <hostname> restarts the smpd on <hostname>

-shutdown shutsdown the local smpd

-shutdown <hostname> shuts down the smpd on <hostname>

-smpdfile <filename> allows you to specify a configuration file other than the

default.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Gforker

只在单节点机器上生成进程的PM，即用于实验和调试。

1. Slurm

外部PM，MPICH2中没有，但提供配置选项引入，--with-pmi=slurm –with-pm=no。

# OpenMPI

[dwt@node0 software]$ tar zxvf openmpi-1.6.2.tar.gz

[dwt@node0 software]$cd openmpi

[dwt@node0 openmpi-1.6.2]$ ./configure --prefix=/home/dwt/software/openmpi

[dwt@node0 openmpi-1.6.2]$ make all

[dwt@node0 openmpi-1.6.2]$ make install

[dwt@node0 openmpi-1.6.2]$ cd examples/

[dwt@node0 examples]$ make

**问题1：MPI\_Allreduce出错**

**btl\_tcp\_endpoint.c:638:mca\_btl\_tcp\_endpoint\_complete\_connect] connect() Connection refused**

mpirun默认调用网桥的网卡接口，导致通信失败

mpirun -np 2 -host node0,node1 **--mca btl\_tcp\_if\_include eth0**