高性能计算集群（PC Cluster）用户指南

大气科学系　应越

Version 2.1　2009-06

目录

第一章：认识cluster

1. 什么是cluster系统

cluster一般由一台主机（master）和多台节点机（node）构成，是一种松散耦合的计算节点集合。为用户提供网络服务或应用程序的单一客户视图，同时提供接近容错机的故障恢复能力。通常cluster 的每台机器通过相应的硬件及软件互连，每个群集节点都是运行其自己进程的独立服务器。这些进程可以彼此通信，对网络客户机来说就像是形成了一个单一系统，协同起来向用户提供应用程序、系统资源和数据。cluster概念的提出在70年代主要是为了进行一些大运算量的科学计算。随着网络的发展，之后的cluster系统还被用作网络服务器，发挥其故障恢复和均衡负载的能力。

使用PC机构建cluster的好处在于开发成本低，而且由于每台节点机都是普通的PC机，在某一台机器发生故障的时候，可以方便的进行维护，而不影响整个系统的运行。

大气科学系的cluster系统，由16台64位的PC机组成。其中一台主机（master），15台节点机（node01~node15）。这16台机器每台有两个4核的CPU，也就是说每个节点上可以同时提供8个CPU。操作系统使用的是CentOS的Linux 发行版。图1为大气科学系cluster目前的结构。其中console和c0101~c0107是大气系早期的cluster系统，节点安装的是RedHat的Linux发行版，precluster 曾经作为门户机，目前已经更新为CentOS的操作系统。

登录master的IP地址为162.105.245.3，这个地址由于物理大楼的IP变动比较频繁，所以可能会时不时改变，而precluster 的IP地址162.105.245.238则比较稳定。这两个地址目前都可以从校外访问。

cluster的应用主要集中在并行计算上。虽然单个节点的单CPU运算效率比普通的笔记本或是台式机都高很多，但是cluster当初被设计出来就是为了进行多CPU协同运算的，而不是仅仅为了提高单CPU的运算效率。所以我们鼓励用户在cluster上进行并行计算，而把一些单CPU也能解决的工作

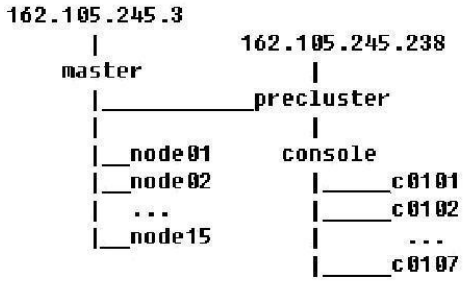

Figure 1: 大气科学系cluster结构

交给自己的PC机完成。

由于master负担了在node**间传递文件和用户信息的重要任务，所以我们应该尽量不要用master主机长时间运行程序，尤其是占用CPU较多的程序，以免进程占用太多CPU而影响其他用户的登陆和文件的传输。

2. linux操作系统

Linux是开源的操作系统，由内核和外部模块构成核心功能，linux上的软件运行以后台进程的方式进行。软件源代码由编译器编译成可执行文件（bin）存放在文件系统中，供用户调用执行。对于维持系统基本功能的服务（service），比如httpd，通常以守护进程（daemon）的方式开机后自动在后台执行。用户同系统的交互由SHELL来完成，这有点类似Windows DOS系统的命令行。用户登录服务器后，通过在SHELL中输入命令来进行操作。

linux的用户分为两种：超级用户（root）和普通用户。root用户拥有所有的权限，普通用户的权限在帐号被创建的时候可以进行相应的设置。

linux系统中的所有文件都被赋有一定的属性，这些属性包括拥有这个文件的用户（user）、组（group）、读写运行的访问权限、最近修改的时间等。其中访问权限的功能非常强大。可以说linux系统的安全就依赖于这样一套严备的体系。

访问权限在linux系统中由一个10位的字符串表示，第一位表示文件的类别：-表示普通文件（file）；d表示文件目录（directory）；l表示链接（symbolic link）。后面的9位分为3组rwx，第一组为文件所有者的访问权限，第二组为文件所有者所在群组的访问权限，第三组为其他用户的访问权限。每组的3个字母：r代表可读权限（readable）；w代表可写权限（wriatable）；x代表可执行权限（executable）。例如：

```
[yingyue@master:~]# ls -l
total 3
-rw-------   2 yingyue dataop 4096 May 17  2008 demo.txt
-rwxrwx---   1 yingyue dataop 4096 May 17  2008 do.exe
drwxr-xr-x   7 yingyue dataop 4096 May 17  2008 home
lrwxrwxrwx   1 yingyue dataop    8 May 17  2008 link -> home/
[yingyue@master:~]#
```

从上例我们可以看到，用户yingyue隶属于dataop用户组，其家目录下有一个home文件目录，两个普通文件。demo.txt 文件只能被yingyue读写，并且不能被执行（rw-）。do.exe文件可以被yingyue以及所有隶属于dataop的用户读写以及执行，但是不能被其他用户读写执行。在cluster上，用户可以设置自己家目录中的文件的访问权限，而对别的用户的文件的访问，则

根据权限设置的不同而不同。

另一个linux操作系统的特点是链接（symbolic link），指向一个链接的文件路径会被自动定向到源文件的位置。比如上面例子中link为一个指向home目录的链接。

cluster的主机和节点之间的文件共享是通过autofs服务实现的。在/etc/auto.misc里定义了本地机器挂载的网络文件目录。/etc/exports里定义了别的机器能够挂载的本机的目录。挂载的文件目录在/misc里可以找到。cluster的机器为了管理方便，将/misc下的目录链接到了/mnt下。用户的信息由master通过yp服务统一管理，每台节点机的/home都挂载为master机器上的/home。用户自己家目录的实际存放点是散布在节点机上的，在/home下链接到实际地点。

在master的/usr/local上安装了的软件，用户可以通过修改PATH环境变量直接调用。附录中列出了目前安装的软件列表。

第二章：使用cluster

当管理员向你提供了用户名和密码后，这表示你已经获取了访问和利用cluster上计算资源的途径。为了展开cluster上的科研工作，我们需要做一些准备工作。

1. 本地准备工作

为了登录cluster，在本地的PC机上需要安装链接服务器SHELL的client程序。对于windows用户，可以使用的软件有:

- SSH Secure Shell Client
  下载地址：http://www.atmos.pku.edu.cn/download/SSHSecureShellClient-3.2.3.exe
- SecureCRT
  下载地址：http://www.atmos.pku.edu.cn/download/securecrt612.exe
- Putty
  下载地址：http://www.atmos.pku.edu.cn/download/putty.exe
- 其他任何附带ssh控制台的软件，比如Ultra Edit等。

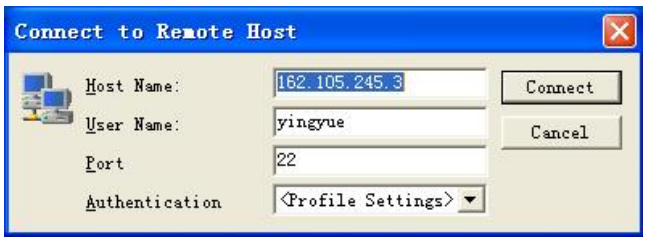

Figure 2: SSH Secure Shell Client登录设置

对于linux用户，可以直接使用系统命令行中的ssh服务：

```
[user@redhat:~]$ ssh yingyue@162.105.245.3
yingyue@162.105.245.3's password:
Last login: Fri Dec  5 16:01:25 2008 from 162.105.139.33
[yingyue@master:~]#
```

纯字符的SHELL命令行通常已经能够满足大多数的调试运行软件的用户。如果用户需要获取cluster的图形界面（X11 tunnelling），还需要安装X window client软件，目前windows下可使用的软件有：

- XWin32
- XManager

2. 服务器端

当准备好链接服务器的client软件之后，最重要的事情是用passwd命令改自己的密码，这是为了保证你的资源安全。当遇到问题需要管理员解决时也不需要提供自己的密码，因为他有办法越过你的权限。你的密码应该只有你一个人知道。

修改密码的方法如下：

```
[yingyue@master:~]# yppasswd
Changing NIS account information for yingyue on master.
Please enter old password:
Changing NIS password for yingyue on master.
Please enter new password:
Please retype new password:
The NIS password has been changed on master.
[yingyue@master:~]#
```

linux系统中的密码在键入时不会显示。注意不能使用passwd命令，因为那样修改后的密码只能在你运行passwd 命令的那台机器上生效。cluster上使用yppasswd命令作用是通过yp服务使修改后的密码被传递到各个节点。

修改完密码后，可以管理一下自己的家目录，linux系统中，当登录用户为yingyue时，～ 、~yingyue 和/home/yingyue 是等效的家目录路径。在家目录中，一般会有一个.cshrc文件存放csh的环境变量。csh是SHELL的一种，其语法比较接近C语言，linux的SHELL还有sh、bash、tcsh等多种，其中bash的使用比较广泛，对应的配置文件是.bashrc。目前cluster上的用户默认SHELL是csh，如果你需要改变这一设置请联系管理员。关于.cshrc文件的详细介绍请参考下一部分内容。

登录cluster后，可以通过rsh命令切换到各个节点，比如从master上rsh登录到node02节点：

```
[yingyue@master:~]# rsh node01
Last login: Tue Nov 25 13:14:03 from master
[yingyue@node01:~]#
```

至此，可以在家目录中建立目录，存放要运行的软件和数据。值得注意的是，如果软件和数据在cluster上已经存在，可以直接调用而不用拷贝到家目录中，这样可以节省硬盘空间的开支。

在平时的使用中，维护自己的家目录是非常重要的，对于软件的编译和运行，一个有条有理的目录能够帮助你顺利完成自己的计算任务。所以我们有必要熟悉linux操作系统的命令。

第三章：linux常用命令

linux操作系统自带的系统命令有很多，然而常用的只有不超过30个。这些命令大致分为文件操作和进程管理两大类。

1. 文件操作

- man [command]
  查看command命令的说明文档（manual page）
- ls -[options] [directory]
  列出目录里的文件，有兴趣的话可以研究一下ls的选项。常用的有-l -a -t等
- cd [directory]
  进入文件夹（不加目录名则默认进入你的家目录）
- pwd
  显示当前所在目录
- rm [files] 删除文件（删除目录需要加-r选项，强制删除用-f）
- cp [source] [target]
  复制文件
- mv [source] [target]
  移动文件（也可以理解为改名）
- touch [filename]
  新建名为filename的文本文档
- mkdir [-p] [directory]
  新建文件夹（-p为建立整个路径）
- ln [-s] [path] [link]
  建立链接（-s为建立软链接）
- cat [textfile]
  显示文本文档的内容
- grep 'content' [file]
  在file中查找有content的行
- sed,awk,cut...
  字符串处理程序
- chown [user.group] [file]
  修改file的所有人和群组
- chmod 755 [-R] [file]
  改变file的访问权限，755三个数字为三组访问权限的加权值。
  r=4，w=2，x=1。755代表的意思是-rwxr-xr-x。又比如644的意思是-r-xr–r–等等。
- tar zxvf [*.tar.gz]
  解压缩文件包，z/j=gunzip/bz2格式，c/x=压缩/解压缩
- find -name [filename]
  在当前文件夹搜索名为filename的文件,有比较多的高级选项
- locate [file]
  快速查找定位文件，只能搜文件名

- file [file]
  查看file的文件类型

- vi
  功能强大的文本编辑工具:
  i　进入编辑模式
  Esc　退出编辑模式
  r　修改单个字符
  :w　保存
  :q[!]　（放弃修改）退出
  :h　帮助
  /string　搜索string
  :2 co 4　将第2行拷贝到第4行

- `[command] > outfile`
  将command命令的执行结果写入到outfile文本文件中

- `&`　在后台执行程序

- `[command1] | [command2]`
  把command1执行的结果作为输入送到command2中执行

2. 环境变量

linux系统的环境变量的作用在于他们定义了应用程序需要多次调用的值，比如：系统文件的路径、IP地址等。定义系统变量可以方便程序获得所需的值，而不必每次都重新定义。

在csh中，定义系统变量的方法是：`setenv NAME value`
或者：`set name=value`
不同的shell使用的定义语句不同，比如bash的语法是：`export NAME=value`
查看系统变量的值，可以用`echo $NAME`。

在csh中用setenv命令定义环境变量，这个变量在csh被关闭之前有效。为了让环境变量永久被记住，则需要将它写入.cshrc文件（对于bash相应的文件是.bashrc）系统在打开一个csh的同时，会自动加载.cshrc中定义的变量。改动.cshrc内容后，需执行source .cshrc或重新登录csh才能生效。让我们来看一个.cshrc实例：

```
1  # .cshrc
2
3  setenv DISPLAY 162.105.39.170:0.0
4  # User specific aliases and functions
5  alias rm 'rm -i'
6  alias cp 'cp -i'
7  alias mv 'mv -i'
8  alias l 'ls -l'
9
10 setenv PATH "/bin:/usr/bin:/usr/local/mpich/bin:
11 /usr/local/ncarg/bin:/usr/local/bin:/usr/local/sbin:
12 /usr/sbin:/sbin:/usr/local/netcdf/bin"
13 setenv PATH "/home/matlab/bin:$PATH"
14 setenv PGI /usr/local/pgi
15 setenv LD_LIBRARY_PATH $PGI/linux86-64/6.1/lib
16 setenv LM_LICENSE_FILE $PGI/license.dat
```

```
17 setenv PATH "$PGI/linux86-64/6.1/bin:$PATH"
18
19 setenv GRAPHCAP /usr/X11R6/include/X11
20 setenv TERM xterm-color
21 setenv NCARG_ROOT /usr/local/ncarg
22 setenv NCARG    /usr/local/ncarg
23 setenv NCARG_NCARG $NCARG/lib/ncarg
24 setenv NCL_COMMAND $NCARG/bin/ncl
25 setenv NCARG_RANGS $NCARG_NCARG/rangs
26 setenv PATH "$NCARG_ROOT/bin:$PATH"
27
28 setenv NETCDF /usr/local/netcdf
29
30 setenv MPICH /usr/local/mpich1
31 setenv PATH "/usr/local/mpich1/bin:$PATH"
32
33 setenv CC pgcc
34 setenv FC pgf90
35 setenv F77 pgf77
36 setenv F90 pgf90
37
38 set prompt="\[`id -nu`@`hostname -s`:%B%~%b\]\#\ "
```

alias rm 'rm -i'的含义是，将rm命令重指向rm -i命令，用户在键入rm命令后，系统执行rm -i。这样使得rm命令执行时始终会有提示信息，防止误删文件。

PATH是最重要的一个环境变量，它的作用是存放可执行命令路径，当你在shell提示符后键入一个命令后，linux会到PATH指定的路径去查找相对应的可执行文件，找到后执行它。所以如果你要调用的命令路径不在PATH中，就得每次都在命令前加上绝对路径才能正常调用。cluster上常用的命令有pgi的系列编译器、mpich并行程序以及一些常用软件里的命令，它们的路径都被加进了PATH变量里。PATH变量的路径用":"隔开。例如，.cshrc文件第13行表示在PATH变量中加入matlab的bin路径。

prompt的作用是改变命令行提示符的表达形式，也就是你在打开一个系统终端（terminal）时系统给你的提示语句。可以修改它让提示符包含更多的信息：上面的例子里的prompt变量让我们的系统提示符变成这样：

```
[user@master:~]#
```

prompt变量只是为了命令行的美观而修改的，你完全可以set prompt=": "，这样也能正常工作，所以prompt的具体设置这里就不赘述了。

3. 进程管理

在平时的使用中，学会进程管理是非常有必要的，因为调试程序的过程中往往会遇到程序非正常退出或失去响应，甚至死循环等现象。这是必须查看程序占用的进程，进行适当的操作。

linux系统的进程分前台和后台两种，当你在shell中输入程序名直接执行，在执行过程中下一个提示符不出现，则程序在前台执行，比如:

```
[yingyue@master:WRF/WRFV2/run]# ./wrf.exe
```

此时命令行直到wrf.exe执行完毕才能键入新命令。wrf.exe在前台运行。

另一种程序的运行方式是后台运行。同样以wrf.exe为例：

```
[yingyue@master:WRF/WRFV2/run]# ./wrf.exe &
```

```
[1] 13242
[yingyue@master:WRF/WRFV2/run]#
```

此时，wrf.exe已经在后台开始运行。进程号(pid)为13242。如果要将wrf.exe的输出结果保存到文本文件的话，可以执行：`wrf.exe > wrf.out`。在后台运行，则执行：`wrf.exe >& wrf.out &`。

查看程序的运行情况可以用ps和top命令:

ps命令是用来查看这一时刻系统上正在运行的进程，用户可以以不同的显示方式来查看进程，从而获取相应的信息。显示系统上正在运行的所有进程，可以使用ps -e，或者ps aux。如果要查看某一程序的进程信息，以wrf.exe为例，可以使用pa aux | grep wrf.exe，这样系统会返回当前所有正在运行的wrf.exe程序。

ps命令返回的是一个时刻的进程信息，如果用户希望动态地监视进程的话（类似于windows中任务管理器的进程模式），则可以选择top命令。top命令执行的结果如下：

```
top - 19:21:28 up 27 days, 10:52,  7 users,  load average: 1.34, 1.11, 1.03
Tasks: 279 total,   3 running, 271 sleeping,   0 stopped,   5 zombie
Cpu(s): 24.1% us,  1.1% sy,  0.0% ni, 74.6% id,  0.0% wa,  0.0% hi,  0.1% si
Mem:   8149796k total,  5787752k used,  2362044k free,    64680k buffers
Swap:  8385888k total,    81776k used,  8304112k free,  5105880k cached

  PID USER      PR  NI %CPU    TIME+  %MEM  VIRT  RES  SHR S COMMAND
 6434 yingyue   25   0  100 218:03.03  0.0 69548 3532 2088 R vim
 3472 yingyue   25   0  100   0:09.01  2.5  295m 200m 4944 R wrf.exe
 3434 ccli      16   0    1   0:00.29  0.0 39460 3064 1228 S sshd
32030 apache    15   0    0   0:00.01  0.1  159m 4508 2932 S httpd
    1 root      16   0    0   1:04.44  0.0  4756  324  292 S init
    2 root      RT   0    0   0:07.63  0.0     0    0    0 S migration/0
    3 root      34  19    0   0:00.69  0.0     0    0    0 S ksoftirqd/0
    4 root      RT   0    0   0:03.57  0.0     0    0    0 S migration/1
    5 root      34  19    0   0:00.14  0.0     0    0    0 S ksoftirqd/1
...
```

在top的界面中可以进行以下操作：按u键，键入用户名，可以只显示某用户的进程；按d键，键入秒数，可以修改top刷新显示的时间间隔；按i键，可以只显示活跃的进程。

有关ps和top命令的更多参数，可以查阅操作手册：man ps和man top。

程序执行完毕后，在shell中会出现如下提示：

```
[1]     Done                          wrf.exe
```

表示wrf.exe已经执行完毕。

如果程序在执行的过程中遇到问题，无法正常中止，可以用killall命令来强行中止。例如：killall wrf.exe。也可以用kill命令直接删除进程，使用ps aux | grep wrf.exe查看wrf.exe程序的进程号(pid)：

```
[yingyue@master:~/WRF/WRFV2/run]# ps aux |grep wrf.exe
yingyue   13242  102  2.5 303084 205108 pts/1 R    20:04   0:18 ./wrf.exe
yingyue   13243  0.0  0.0 51092  684 pts/1    S+   20:04   0:00 grep wrf.exe
```

可见wrf.exe的进程号是13242，执行kill 13242来中止它。

4. 脚本

使用脚本，可以使繁琐的工作变得简单，也方便管理自己的程序。概括地说，脚本是shell中的一个命令集合，可以将多个命令作为一个单一文件

执行。在日常的工作中我们经常会遇到这样的情况，完成一项工作需要执行一连串的命令，并且在整个过程中需要根据结果的不同做相应的判断，脚本的出现使得我们不必自己一次次重复复杂的操作，而是将规则记录下来让计算机去为我们操作。

掌握脚本的使用，最重要的是理解变量、赋值和条件判断。以csh为例，赋值的方法是：set var="string"。var是变量名，string是一个字符串。调用var的值的方法是在它的前面加上$，例如$var或${var}。在字符串中混用变量的情形下，为了避免变量后紧跟字符串造成的歧义，应使用第二种表示方法。例如：

```
set str="Satur"
echo Today is ${str}day!
```

脚本的运行结果是：Today is Saturday!

由于脚本是基于命令行输入输出的编程语言，所有的操作基本以字符串为基础，所以变量的类型只能是字符串。值得一提的是，在脚本中可以方便地将命令执行的输出作为字符串赋给一个变量，只需使用‘（~键）就可以了。例如：set currentdate=‘date‘，这样便将date命令的输出赋给了currentdate变量。

csh中的条件判断和C语言比较类似，同时还增加了文件系统的判断功能，例如：

```
if ( -f $filepath ) then
  rm $filepath
end if
```

上例查找$filepath是否存在，若存在，则删除这个文件。-f判断文件是否存在，-d判断文件夹是否存在。

由于所有的变量都是字符串，所以对数的运算也是通过字符串的命令完成的。Linux提供了expr命令来弥补脚本对于数值运算支持的不足。

下面举一个简单的例子（定期下载天气图的脚本download.csh）：

```
1  #!/bin/csh -f
2  set date=‘date +%Y%m%d‘
3  set hour=‘date +%H‘
4  set dir=/mnt/storagedata1/data/synoptic_chart/KMA
5
6  foreach hr ( 0 3 6 9 12 15 18 21 )
7    set offset=‘expr 8 + \( $hour - 8 \) % 3 + $hr‘
8    set ccyymmddhh= \
9        ‘date -d ${offset}\ hours\ ago\ ${date}\ ${hour}:00 +%Y%m%d%H‘
10   set ccyy=‘echo $ccyymmddhh |cut -c1-4‘
11   set mm=‘echo $ccyymmddhh |cut -c5-6‘
12   set dd=‘echo $ccyymmddhh |cut -c7-8‘
13   set hh=‘echo $ccyymmddhh |cut -c9-10‘
14   if ( ! ( -d $dir/$ccyy$mm ) ) then
15     mkdir -p $dir/$ccyy$mm
16   endif
17   cd $dir/$ccyy$mm
18   echo kma_sfc3_${ccyy}${mm}${dd}_${hh}.png
19   if (!(-f $dir/$ccyy$mm/kma_sfc3_${ccyy}${mm}${dd}_${hh}.png )) then
20     wget http://203.247.66.10/cht/img/sfc3_${ccyy}${mm}${dd}${hh}.png
21   mv sfc3_${ccyy}${mm}${dd}${hh}.png \
22      kma_sfc3_${ccyy}${mm}${dd}_${hh}.png
```

```
23   endif
24 end
```

第1行说明了这个文本文件是一个csh脚本；2、3行将当前的时间赋给变量；第6行定义了一个循环，对每一个时次进行一系列操作；第14行判断文件夹是否存在；第19行判断文件是否存在，若不存在则下载文件。

执行csh脚本前，需要将文件chmod为可执行文件：

```
chmod 755 download.csh
```

对于字符串处理，Linux系统提供的awk和sed命令具有更加完备和强悍的功能。有兴趣的读者可以自行查阅它们的说明文档。这两个强大的工具使得复杂的字符串操作成为可能，所以在脚本中经常被用到。

第四章：软件

Linux系统下的软件安装与Windows系统中非常不同，Windows系统中我们习惯于运行setup.exe文件，其实它所做的工作是将自身压缩的可执行文件解压并拷贝到系统中，并在注册表总留下相关的记录，使得软件能够正常运行。在Linux系统中没有注册表，另外由于开源软件的流通，软件经常以代码包的形式被下载使用。这样做的一个好处是软件包所占空间非常小，缺点是在安装前需要编译。

1. 编译器

软件的代码通常由C语言、Fortran语言等写成，Linux也提供了相应的编译器，例如GNU的gcc，gfortran，Intel 的f90等。当然也有不少的软件公司提供自己研发的编译器，比如我们cluster上使用的是PGI的编译器（pgcc，pgf90等）。

在Linux系统中编译源文件非常简单，只需要执行命令[compiler] -flag [sourcefile] -o [executive]。compiler为编译器名，flag为编译过程中的参数设置，sourcefile为源代码文件，executive为编译成的可执行文件，默认的文件是a.out。例如：

```
pgf90 -Mfree -byteswapio test.f -o test.exe
```

该命令用pgf90编译器编译test.f文件，编译过程中使用了free和byteswapio选项，将生成的可执行文件命名为test.exe。

编译器的选项设置是否正确，影响到编译是否能顺利完成。这些选项包括系统类型的选择、内存的使用、需要用到的链接库等。

2. 库

熟悉编程的读者应该对库的概念有所了解，Linux系统中编译源代码指定链接库文件的方法是（以pgi编译器为例）：

```
pgf90 -L/usr/lib -ldemo test.f
```

这样指定了在编译test.f的过程中链接demo库文件libdemo.so，该文件位于/usr/lib中。

cluster中的许多软件使用了netcdf、ncarg等链接库。再来看一个更复杂的例子：

```
pgf90 -I -byteswapio -L/usr/local/ncarg/lib -L/usr/X11R6/lib64
      -lncarg -lncarg_gks -lncarg_c -lX11
      -L/usr/lib64 -lg2c test.f
```

以上的编译选项中，链接了/usr/local/ncarg/lib，/usr/X11R6/lib64，/usr/lib64中的库文件：libncarg.a，libg2c.so等。

3. Makefile

由于软件包往往是由大量的源码文件组成，它们之间又有着复杂的依赖关系，如果依次单个进行编译的话，会非常的耗时耗力，所以Linux提供了make机制来处理复杂的软件编译过程。在软件包的各目录中，都有一个Makefile文件，记载了该文件夹中的源码按什么样的规则来编译。在软件的根目录中，同样有一个Makefile 记录软件的作者提供的可能的编译方式。Makefile中记录了编译器名、编译器使用的选项以及源代码被编译的先后顺序等。因此，我们在软件的编译过程中只需要修改Makefile中的相应项就可以了。

4. 编译软件的步骤

软件编译的第一步，始终是阅读readme文件，因为软件的作者会在里面详细地介绍软件编译安装的过程以及可能遇到的问题。

为了方便用户的编译，许多软件的作者提供了configuration这一步，有点类似于windows软件“下一步”的风格。运行configure脚本进行用户交互，根据得到的选择生成合适的Makefile，使得用户不用亲自研究Makefile的语法。configure脚本在执行完后，会生成一个配置文件，Makefile中会调用这个文件，使得设置生效。

在正确修改Makefile之后，在软件的根目录下执行make命令，开始编译。

如果在编译中遇到了错误，会在输出文本中体现出来。常见的错误有：使用了错误的编译模式、链接的库文件没有找到、系统兼容性等。

5. 并行计算任务

cluster的主要作用是进行并行计算，并行计算的软件与一般的软件较为不同。使用pgf90命令直接编译的可执行文件，在执行的过程中只会占用一个进程。进行并行计算需要将软件按照并行计算的模式进行编译。对于软件的使用者来说，并不关心其具体的实现方法。并行计算软件的作者通常已经将这些编译模式设计好，记录在Makefile 中供用户使用。

cluster上使用的并行计算环境是MPI(/usr/local/mpich1)，其编译软件的命令是mpif90等。用户可以在软件包中找到相应的设置。

在cluster提交并行作业需使用PBS作业管理系统，PBS系统会根据cluster上的计算资源使用情况对提交的任务进行排队，确保cpu在不被重复占用的情况下最大化使用。用户提交任务方法如下：

假设任务根目录为`/home/yingyue/wrftest/`，可执行并行程序为wrf.exe。我们在`/home/yingyue/wrftest/`下写一个脚本用来存放提交给pbs系统的命令，脚本可以任意命名，这里以myrun为例，其内容：

```
1 #!/bin/csh
2 #PBS -N myjob
3 #PBS -o output
4 #PBS -e error
5 #PBS -l nodes=1:ppn=8
6 cd /home/yingyue/wrftest/
7 mpirun -np 8 -machinefile $PBS_NODEFILE ./wrf.exe
```

在master上，通过命令qsub myrun提交任务到PBS系统。系统会返回一个任务标识号，类似于1944.master。

注意到脚本中第1行是告诉系统这是个csh脚本。第2-5行是由PBS系统读取的参数，注意#PBS这四个字符告诉PBS系统该行是提交任务的参数，在脚本中#PBS行中的参数也可以在qsub命令后提交，比如

qsub -N myjob -l nodes=1:ppn=8 myrun。

任务的参数有很多，-N指定任务名、-o和-e指定一般输出和错误输出的文件名、-l指定任务占用资源。用户可以指定使用CPU的数量，也可以具体指定使用哪些CPU。指定CPU数量的方法是设置nodes=N:ppn=M，其中N是节点的数量，M是每个节点上占用CPU的数量，也就是总共使用M*N 个CPU，PBS系统会自动寻找到合适的CPU分配这些作业。如果用户需指定使用的节点，可以设置nodes=A:ppn=M+B:ppn=M ，其中A,B是节点的名称，比如：nodes=node01:ppn=8+node02:ppn=8 。同样，PBS会在这些节点空闲时分配用户的作业。

某些时候，ppn的值可以不指定，比如nodes=node01，这时虽然作业可能占用了node01 上的8个进程，但在PBS系统中只记录了nodes=node01:ppn=1，node01对于别的作业来说是free 的，只有当nodes=node01:ppn=8时，node01 才是job-exclusive 的。所以指定ppn 的值是效率最高的做法，因为可以防止别的作业被重复提交上来。查看节点机的状态（free/job-exclusive），可以用pbsnodes -a命令。

第6行进入到工作目录，输出文件output和error以及程序运行过程中的输出文件会被创建在该目录。

第7行是执行并行计算程序的命令。-np为使用的cpu数目，这个要和第5行中定义的一致。-machinefile参数后用$PBS_NODEFILE变量指定mpirun的节点文件，这个是PBS系统来完成的任务。有关qsub的更多说明请参考说明文档。

提交完任务后，可通过qstat查看任务状态。cluster系统可以同时提供8*15个CPU进行并行计算。node14和node15被用作每天的一些例行计算任务（数值天气预报等），所以用户可以使用的是总共8*13个CPU。

```
[yingyue@master ~]$ qstat -a

master:
                                                            Req'd  Req'd   Elap
Job ID       Username Queue    Jobname  SessID NDS TSK Memory Time  S Time
------------ -------- -------- -------- ------ --- --- ------ ----- - -----
1857.master  yangda   workq    run       31540   1  --    --    --  R 105:3
1858.master  yangda   workq    run       28355   1  --    --    --  R 105:3
1859.master  yangda   workq    run        6383   1  --    --    --  R 105:3
1860.master  yangda   workq    run       32469   1  --    --    --  R 105:3
...
1926.master  zhujiang workq    runfoam   19722   1  --    --    --  R 13:30
1938.master  daizp    workq    run        6316   3  24    --    --  R 06:00
1939.master  shirleyr workq    run.csh     --    2  --    --    --  Q   --
...
1944.master  yingyue  workq    myrun       --    1  --    --    --  Q   --
```

可以看到，1944.master就是我们刚才提交的wrf.exe任务，现在的状态是Q，即在排队中，等待队列前的任务完成。也可以qstat -f 1944查看任务1944的具体情况。

要删除提交的任务，使用qdel 命令；改变任务属性，使用qalter 命令。对于这些命令的使用方法，说明文档中有详细的解释，这里不再赘述。

`qsub myrun` 程序开始运行后，可以用top命令查看进程占用CPU的情况。如果程序出现错误需要强行停止，可以用killall和kill命令。

第五章：文件传输

master和precluster开通了httpd服务和vsftpd服务，可以供用户进行http和ftp 协议的文件传输。

1. ftp

使用帐号登录ftp，可以访问自己家目录下的文件，并可以执行读写操作。

ftp client软件有许多种，对于windows用户，可以使用IE浏览器自带的ftp协议，也可以使用专门的ftp软件，比如LeapFTP等。ftp登录的方法是：

```
ftp://user:passwd@162.105.245.3:21
```

user为用户在cluster上的用户名，passwd为密码。

linux用户可以使用lftp程序登录cluster进行文件传输。

2. sftp sz scp

除了ftp协议之外，SSH也提供文件传输的功能，比如SSH Secure Shell Client软件在安装后会自带一个SSH Secure File Transfer Client，提供类似ftp的文件传输功能。

对于linux用户，使用scp或sz命令可以方便的进行文件传输。

要将cluster服务器端的文件下载到本地，在本地端命令行键入：

```
scp user@162.105.245.3:/path/to/file /source/dir/.
```

系统会提示输入密码，将服务器端的/path/to/file下载到本地端/source/dir/中。

要将本地文件上传到cluster，在本地命令行键入：

```
scp /path/to/file user@162.105.245.3:/path/to/file
```

系统同样会提示输入密码。

如果知道本地端的IP地址，那么在服务器端用scp传输文件同样可行。

sz命令具有类似scp的功能，用法相对简单，它会将文件服务器上的文件下载到预设的本地目录中。

3. http

如果用户希望文件能在internet上被访问到，并且可以下载，这也是可以办到的。只需在自己的家目录中建立`public_html`文件夹，将相应文件拷贝(cp)或者链接(ln -s)到里面。就可以通过`http://162.105.245.3/~user/`来访问了。

http方式只能用文件下载，并不能将文件上传到cluster。用户可以将自己的文件，比如某个程序的运行结果，公布到网上，方便老师同学查看。

值得注意的是，由于internet可以被任何人访问，也包括搜索引擎，这意味着用户公布的文件可以被internet上所有的用户看到，如果文件中含有隐私信息的话，就不好了。希望读者留意这一点。另外，也不要在`public_html`文件夹中存放大的影音文件、数据资料等，防止被P2P盗链下载软件搜索到，引起集中下载，大量浪费cluster网关的费用。

附录：cluster上安装的软件(/usr/local)

1. netcdf
2. pgi
3. ncarg
4. ncl
5. nco
6. matlab
7. idl
8. hdf
9. mpich
10. hdf
11. ferret
12. grads

后记2008-12

以上指南为笔者根据自己对cluster系统的理解，结合日常管理中用户经常提出的问题写成。第二版修正了第一版中的不少错误，并且增加了一些内容。由于对计算机的钻研较少，对计算机的知识停留在使用的层面，所以难免在叙述上存在误差，望读者指正。

大气科学中的许多研究课题都需要使用模式来模拟大气的状况，所以计算机成为这一学科必不可缺的科研工具，但是在使用这个强大的工具的过程中，遇到技术上的问题时，往往会影响科研的进度。这让不少我的同学感到苦恼。所以我希望成立cluster技术小组，组织起一支技术力量来为科研工作服务。

后记2009-06

为了更好地利用有限的计算资源，笔者一直致力于为用户提供更好的计算环境。这个学期cluster的一个重大改变，是抛弃了用于自行寻找节点、通过mpirun直接提交任务的方式，改用了PBS作业管理系统来对用户的任务进行排队管理。从而提高cluster使用的效率。对于直接提交的任务，有可能重复占用cpu，威胁到任务的运作效率，因此系统定时扫描“非PBS”任务，并将其终止，以便维护排队系统中作业的正常运行。

笔者还为cluster增加了日志系统，记录用户使用cpu的情况：

`http://162.105.245.3/~yingyue/cluster`

我的联系方式：

`hardyying@gmail.com`

蔚蓝空间研究生论坛

`http://groups.google.com/group/azurespace_pku`