



中科院计算所
INSTITUTE OF COMPUTING TECHNOLOGY, CAS

搭建并行系统

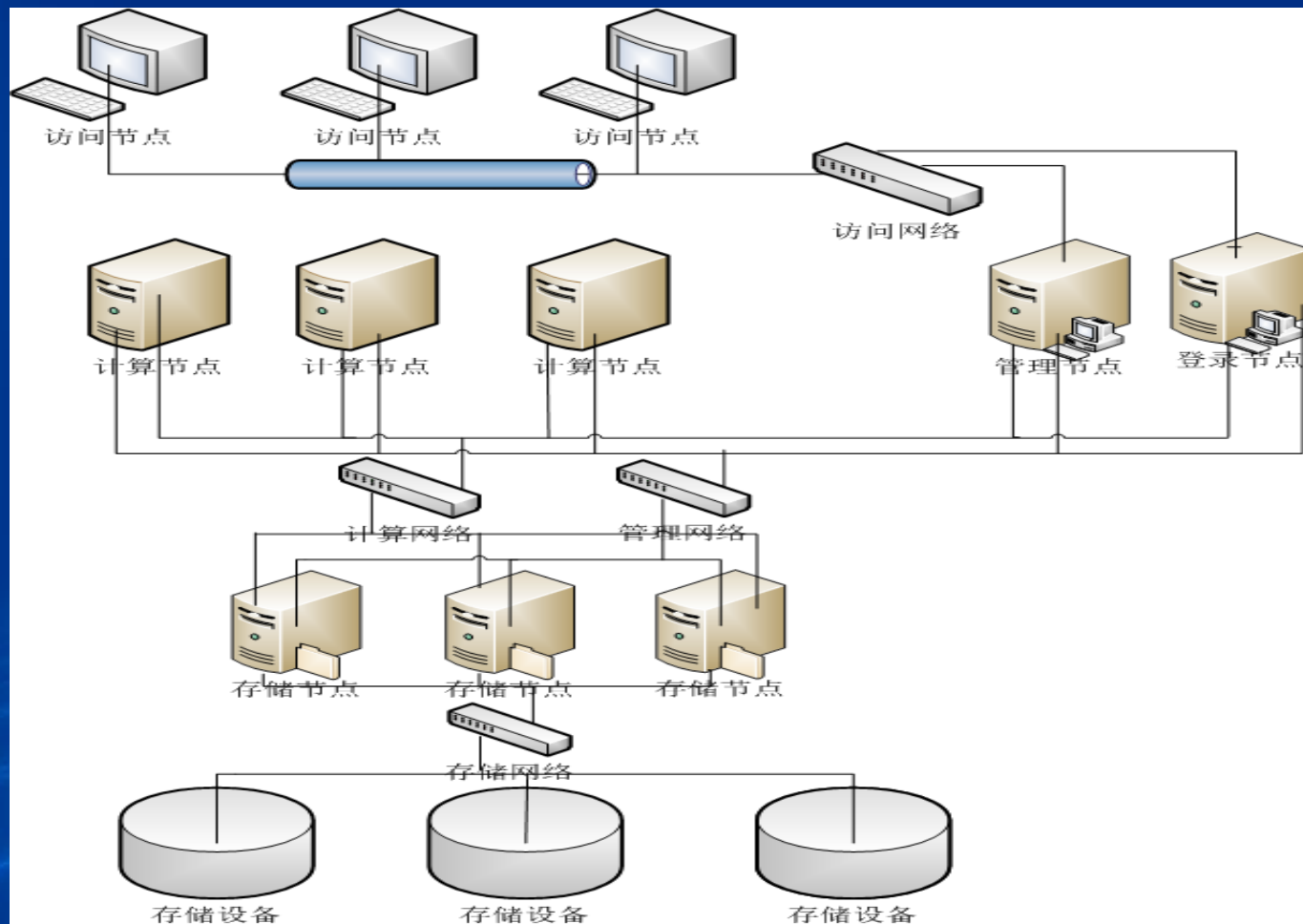
谭光明 博士

tgm@ncic.ac.cn

中国科学院计算技术研究所
国家智能计算机研究开发中心
计算机体系结构国家重点实验室

集群介绍

一个标准集群的架构图



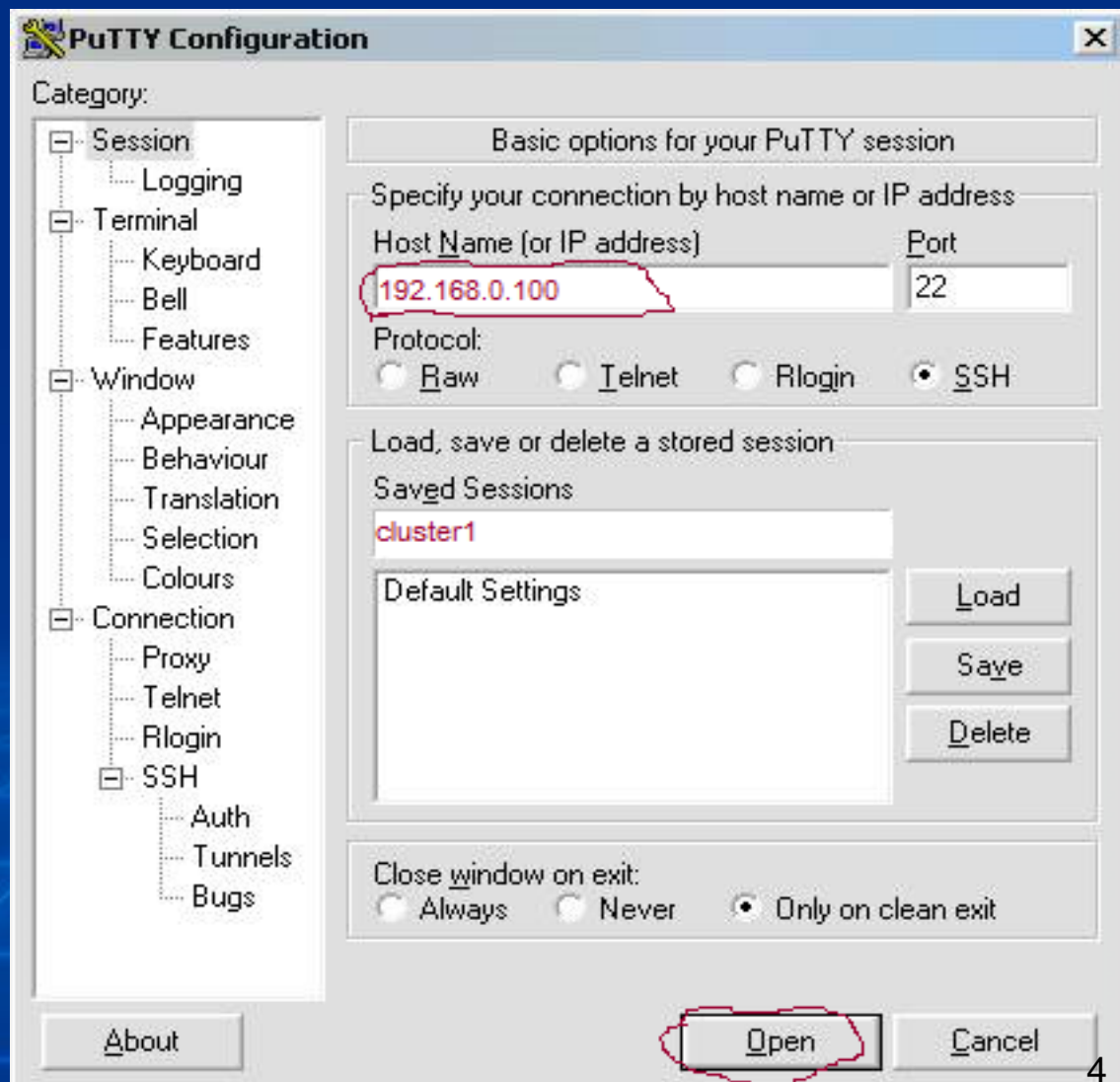
集群工具

- 文本方式
 - Putty
 - SSH secure shell client
- 图形界面
 - VNC
 - xmanager
- 文件传输
 - Ssh secure file transfer
 - ftp客户端



集群工具

- Putty



集群工具

- VNC

在服务器节点上执行

```
node200#vncserver
```

New 'node200:1 (whwen)' desktop is node200:1

记下端口号，如上为1，第一次执行时需要输入密码

vnc配置：

修改`~/.vnc/xstartup`，将`twm` & 修改为`startkde` &

vnc相关命令

```
vncserver -kill :1 #杀死端口号为1的vnc进程
```

```
vncviewer 192.168.0.100:1, #登录IP192.168.0.100，端口号为1 的机器
```

```
vncpasswd #修改vnc密码
```



系统命令

- 节点间相互登录命令

命令: rsh, ssh,

node200#rsh node100

- 节点间拷贝文件命令

命令: rcp, scp, 第一个参数为源文件, 第二个为目标文件
从node200上往node100拷贝, 拷贝目录时加-r

Node200#scp ./file1 node100:/home/test/

从node100上往node200上拷贝, 拷贝目录时加-r

Node200#scp node100:/home/test/file1 ./



系统常用命令

- 集群常用服务开启命令

- 命令: `chkconfig`

- 用法:

- 查询: `chkconfig --list |grep nfs`
 - 添加: `chkconfig --add nfsserver`
 - 删除: `chkconfig -del nfsserver`

- 命令: `service` (大多服务在 `/etc/init.d` 下可找到)

- 用法:

- `Service nfsserver start` #启动 nfsserver
 - `Service nfsserver stop` #停止 nfsserver
 - `Service nfsserver status` #查看 nfsserver 状态



系统常用命令

- 几个特别的集群命令

- 命令: `nprsh`, `pdsh`

- 解释: 多节点并行执行脚本命令

- 用法:

- `nprsh -on node1..80 hostname` #在节点1至80上执行hostname命令

- 命令: `pdsh`

- 解释: 多节点并行执行软件命令

- 用法: `pdsh -R rsh -w node2,node3 hostname`, #在节点 2, 3上执行hostname

```
TC5000:/home1/scripts # nprsh -on node1..10 hostname
Run "rsh NODE hostname" on:  node1 node2 node3 node4 node5 node6 node7 node8 node9 node10
-----OUTPUT-----
node1
node8
node5
node2
node9
node7
node4
node6
node10
node3
```



系统常用命令

- 几个特别的集群命令

- 命令: pestat
- 解释: 在装完调度系统torque之后的脚本命令, 查看整体集群负载情况
- 用法: 管理节点上执行pestat
- 示例

```
[root@node11 ~]# pestat
```

node	state	load	pmem	ncpu	mem	resi	usrs	tasks	jobids/users
node1	busy*	8.03	16029	8	20124	14980	2/1	8	2536 student
node2	busy*	8.01	16029	8	20124	14986	1/1	8	2536 student
node3	busy*	8.05	16029	8	20124	15042	1/1	8	2536 student
node4	busy*	8.30	16029	8	20124	15013	2/1	8	2536 student
node5	free	0.00	16029	8	20124	201	0/0	0	
node6	free	0.00	16029	8	20124	201	0/0	0	
node7	free	0.05	16029	8	20124	209	0/0	0	
node8	free	0.00	16029	8	20124	254	0/0	0	



系统常用命令

- 常用安装软件命令

- 命令: rpm
- 解释: 软件包管理器
- 使用:
 - 安装: `rpm -ivh rsh-server0.1-5.rpm`
 - 卸载: `rpm -e rsh-server0.1-5`
 - 查询: `rpm -qa|grep rsh`
 - 更多命令选项见man
- 命令: tar
- 解释: 文件归档命令, 可压缩
- 使用:
 - 解压: `tar xzf openmpil.3.3.tar.gz;tar jxf openmpil.3.3.tar.bz`
 - 归档: `tar zcf openmpi.tar.gz openmpil.3.3`



并行机群系统调试

2016/8/23



中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

RSH的配置

- 配置启动文件

开启rlogin rsh rexec 服务

```
Chkconfig --level 35 rlogin on
```

```
Chkconfig --level 35 rsh on
```

```
Chkconfig --level 35 rexec on
```

RSH的配置

- #vi /etc/hosts

把第一行的主机名删掉并加入曙光服务器的所有节点IP地址和主机名

- 如：原文件为：

```
127.0.0.1    node1    localhost.localdomain  localhost
.....
```

- 修改后应为：

```
127.0.0.1    localhost
192.168.0.1   node1
192.168.0.2   node2
.....
```

```
192.168.0.x   nodex
.....
```

- #vi /root/.rhosts

把机器的所有节点的主机名加入

如：node1

node2

.....

nodeX

- X：为节点的数量

RSH的配置

Cat /root/.rhosts >/etc/hosts.equiv

机群中每台机器需要三个相同的文件

/etc/hosts /etc/hosts.equiv /root/.rhosts

vi /etc/securetty

添加

rsh

rlogin

rexec

执行service xinetd restart

测试: rsh nodex(x为节点号) ls -l 测试



SSH的配置

- 在系统中执行ssh nodeX, 在当前用户目录生成.ssh/目录,
- 切换到相应用户 su username
- 执行ssh-keygen -t rsa, 一路默认回车, 生成id_rsa.pub文件
- cat id_rsa.pub >>authorized_keys
- 将机群中每个机器生成的id_rsa.pub中内容全部cat到authorized_keys文件中, 然后保证每台都有同一个authorized_keys文件, 即可互相ssh不用输入密码



机群常用服务配置

NFS服务

- NFS(network file system)
- NFS 配置:

服务端server配置

1. 开启nfs 服务: `chkconfig -level 35 nfs on`
`/etc/init.d/nfs start`
2. 编辑/etc/exports 假设将/public共享给网络
`/public *〈不限制用户〉(rw,no_root_squash 〈信任客户端〉,async 〈先写入内存〉)`
3. 执行`exportfs -a`
执行`exportfs` , 可以看到 `/public 〈world〉`

客户端配置:

1. 建/public目录 `mkdir /public`
2. 添加mount server:/public /public 到
`/etc/rc.local(rhel),/etc/init.d/after.local(suse)`
3. 执行`/etc/rc.local(rhel),/etc/init.d/after.local(suse)`
4. `df` 命令看mount 情况

NIS

- NIS : Network Information Services 对系统信息进行集中管理
- 服务端配置
- 启动NIS服务

```
Chkconfig -level 35 ypserv on
```

```
Chkconfig -level 35 yppasswdd on
```

```
Chkconfig -level 35 ypxfrd on
```

- 编辑/etc/sysconfig/network
添加 NISDOMAIN=cluster (这里可以用你喜欢的域名)
立刻使上述设置生效, 运行: # domainname cluster

- 编辑 /var/yp/Makefile
找到all: passwd group 这一行, 修改如下:
all: passwd group

- 启动ypserv服务
service ypserv start

- 初始化NIS数据库, 运行:
#/usr/lib/yp/ypinit -m
注意: 1) 若以上配置有问题, 会显示出错。
2) 提示增加hosts, 按[CTRL-D]



NIS

- 若上部没有错误，重新启动ypserv
service ypserv restart
- 启动密码进程：
service yppasswdd start
- 检查进程是否真的运行，运行：
ps auxf | grep yp
- 检查日志，确定NIS服务是否真的正常，运行：
tail /var/log/messages
- 将用户目录/home做NFS网络输出
#Vi /etc/exports
添加/home *(rw,no_root_squash,async)
#exportfs -a



NIS

- NIS客户端
- 开启NIS服务:

```
#chkconfig -level 35 yppasswdd on
```

```
#chkconfig -level 35 ypbind on
```

- 编辑/etc/sysconfig/network
添加 NISDOMAIN=cluster (与服务端一致)
- 编辑 /etc/nsswitch.conf
- 启动客户端

```
#/etc/init.d/yppasswdd start
```

```
#/etc/init.d/ypbind start
```

- Mount服务端用户目录

```
#mount server:/home /home
```

添加 mount server:/home /home 到客户端/etc/rc.local文件中



NIS

- NIS测试
- 运行 `# ypcat passwd` 看服务器提供了哪些用户
- 在NIS Server上新增一用户

```
useradd test
passwd test
cd /var/yp
执行 make 命令
```

在客户机上用test用户登录
- 注意事项:
 1. NIS Server的自启动: 先启动NIS服务端, 后启动客户端
一定要注意让它自启动, 否则下次开机nis server不能启动的话, 客户端查询不到nis server, 将不能登录
 2. 每增加一个用户, 或者用户修改了密码等, 要在服务器端的var/yp下执行一下make命令。
 3. 新增用户和修改密码一定要在NIS Server上进行, 这样所有的客户端就会自动更新



FTP

- 软件包安装
 - `Rpm -ivh vsftpd-2.0.4-19.11.x86_64.rpm`
 - 配置文件:`/etc/vsftp.conf`
- FTP服务开启:
 - `#Chkconfig --level 35 vsftpd on`
 - `#/etc/init.d/vsftpd restart`
- 增加download用户
 - `#useradd download`
 - `#passwd download` (输入用户密码)
- 设置用户权限为777 (权限全部放开)
 - `#chmod 777 /home/download`
- 客户端登陆:
 - `ftp server`
 - 用户名 `download` , 输入相应密码即可

常用MPI环境安装

- 详细请参考文档《MPI安装》



应用软件调试

软件运行出错后怎么办？

- 随时记录运行日志
 - 执行时将正确和错误输出都导入到一个文件中
 - `mpirun -np $NP -hostfile ./ma a.out >&a$(date).log`
- 软件的输入是否正确
 - 找个benchmark来测验下
 - 版本不匹配
 - 运行核数不正确
- 软件的版本问题？
 - 是否支持并行
 - 支持哪种并行，MPI还是OpenMP, 或者PVM
- 软件的扩展性？
 - 运行进程从小到大试试看，
- 运行命令是否正确？
 - 程序是否需要输入，
 - 不同的MPI不同的执行方式，耦合模式的执行方



集群使用建议

- 使用Linux命令时，可以通过man手册获得相关命令详细使用说明
- 执行命令出错时，可通过输入错误关键词在网络上获得相关信息，（同样的错误别人肯定也经历过）
- 编译软件出现编译器相关错误时考虑换编译器，或者升级或降低软件版本
- 安装软件时，首先查看其手册，如INSTALL, Readme等，然后可查看Makefile或makefile检查其依赖的专业或系统库，确保正确的路径被添加，同时检查makefile依赖的其他文件，确保正确的编译环境。
- 可借鉴别人的成功编译环境，（包括软件版本，编译器，MPI，操作系统等）
- 最后，实在未见过的错误最好去google下



目录

- clusconf 简介
- clusconf 安装与设置
- clusconf 使用
- clussoft 使用

傻瓜式配置机群系统



clusconf 简介

- 一个配置和管理集群的命令行工具，使用他可以方便快速的完成对集群的初始配置和后期管理
- Clusconf 三类操作（ operation ）
 - 集群配置操作，用于集群配置
 - 集群管理操作，用于配置完成后的集群管理维护
 - IPMI操作，使用IPMI相关功能控制集群

clusconf 1.4 changelog

- 1 Adding parallel excuting mode,and same results combine
- 2 Adding ipmi relative operations
 - clusconf --set-ipmi
 - clusconf --ipmi-info
 - clusconf --ipmi-poweron
 - clusconf --ipmi-ipoweron (Intelligent Power On)
 - clusconf --ipmi-poweroff
 - clusconf --ipmi-ipoweroff (Intelligent Power Off)
 -
- 3 Adding operation to display cluster info
 - clusconf --sys-info
- 4 fix bug
 - 1) --sync-file
 - create path if there are no parents path in other nodes
 - 2) check os and warning if all of OS is deferent in the cluster
 - 3) error if tab keys used in /etc/hosts
 - 4) --adduser|--useradd
 - can specify the user uid
 - 5) --sync-time adding the command to /etc/crontab



高性能集群三要素

统一的文件印象

对于普通用户来说，所有节点看到某一个文件都是相同的文件。通过nfs或者并行文件系统实现。

统一的用户印象

通过NIS或同步用户

无密码访问配通

网络全通

或ssh无密码访

clusconf -----使高性能集群三要素实现



Clusconf 使用前要求

- 集群硬件安装到位，线缆连接正常
- 节点操作系统安装正确。
- 网络线缆连接正确，节点IP地址配置正确



Clusconf 安装与配置

■ 安装clusconf

```
1 tar xvf clusconf-1.4.tgz
```

```
2 cd clusconf-1.4
```

```
3 ./install
```

根据提示输入安装路径，默认路径为/opt/clusconf

查看安装路径

载入环境变量

```
4 source /etc/profile.d/clusconf-env.sh
```

```
5 which clusconf查看安装路径
```

■ 配置/etc/hosts 文件

```
127.0.0.1      localhost
```

```
10.10.10.1    node1  comput1
```

```
10.10.10.2    node2  comput2
```

```
.....
```

```
10.10.10.30   node30 comput30
```

```
10.10.10.31   node31 io1
```

```
10.10.10.32   node32 io2
```

```
10.10.10.33   node33 admin1
```

```
12.12.12.1    inode1
```

```
.....
```

```
12.12.12.33   inode33
```

clusconf 配置

- 修改<installdir>/etc/nfs.cfg (如果使用clusconf 配置nfs 文件系统)

```
NFSDIR inode31 /public /public
NFSDIR inode32 /public2 /public2
# NFSDIR node33 /data1 /data1
# NFSDIR inode40 /data2 /data2
BINDHOME /public/home
```

- 修改<install>/etc/ipmi.cfg (如果使用clusconf IPMI相关功能)

# ipmiaddr	hostname	character	ipmiusername	ipmipasswd
11.11.20.1	node1	comput	root	superuser
11.11.20.2	node2	comput	root	superuser
11.11.20.3	node3	io	admin	admin
11.11.20.4	node4	io	root	superuser
11.11.20.5	node5	admin	root	superuser
11.11.20.6	node6	admin	admin	admin

clusconf 使用

■ clusconf [Options(选项)] [Operation (操作)]

■ 选项如下：

- -h|-help|--help 查看帮助
- -p|--prefix 标识节点的前缀，默认值为node
- -n | --snlist标识节点的序号，默认为/etc/hosts 中所有的具有前缀的节点。
- -r 标识节点间交互使用ssh或者rsh，默认是使用ssh
- -s 标识进行操作时使用顺序模式。默认关闭该选项，使用并行模式。

■ 操作分为如下三类操作

- 配置操作，用于集群初期配置
- 管理操作，用于集群管理
- IPMI相关操作，使用IPMI相关功能



clusconf 使用

■ -p 和 -n 的使用

- `clusconf --sync-do hostname`

标识的节点为/etc/hosts 中所有的node为前缀的节点

- `clusconf -p comput --sync-do hostname`

标识的节点为/etc/hosts 中所有的comput为前缀的节点

- `clusconf -n `seq 1 30` --sync-do hostname`

标识的节点为node1-node30

- `clusconf -p inode -n `seq 1 30` `seq 35 50` 55 56 --sync-do hostname`

标识的节点为inode1-inode30, 加上inode35-inode50, 再加上inode55, inode56



clusconf 使用

■ 并行模式与顺序模式

- ❑ -s 打开顺序模式，默认使用并行模式
- ❑ 并行模式：多节点操作时并行执行，并且自动合并相同的执行结果
- ❑ 顺序模式：多节点操作时顺序执行，并且按顺序显示执行结果。

并行模式执行

```
clusconf --sync-do cat /etc/issue  
node[1,3-5]
```

```
-----  
Red Hat Enterprise Linux Server release  
6.2 (Santiago)  
Kernel \r on an \m
```

```
-----  
node2
```

```
-----  
VMware ESX 4.1 (Kandinsky)  
Kernel \r on an \m
```

顺序模式执行

```
clusconf --sync-do cat /etc/issue
```

Excuting Command cat /etc/issue On
the whole Cluster

```
=====node1=====
```

```
Red Hat Enterprise Linux Server release  
6.2 (Santiago)  
Kernel \r on an \m
```

```
=====node2=====
```

```
VMware ESX 4.1 (Kandinsky)  
Kernel \r on an \m
```

```
=====node3=====
```

```
Red Hat Enterprise Linux Server release  
6.2 (Santiago)  
Kernel \r on an \m
```

```
=====node4=====
```

```
Red Hat Enterprise Linux Server release  
6.2 (Santiago)  
Kernel \r on an \m
```

```
=====node5=====
```

```
Red Hat Enterprise Linux Server release  
6.2 (Santiago)  
Kernel \r on an \m
```

Clusconf 配置操作

- 1设置某用户的ssh无密码访问 (--set-ssh username)
 - 按照提示，输入回车即可
- 2设置所有用户的rsh无密码访问 (--set-rsh)
 - 系统会检查是否已经安装了rsh-server rpm 包，如果没有安装，请安装rsh-server rpm 包，然后再执行此条命令
- 3按照集群规范设置服务 (--set-service)
 - 会自动关闭防火墙，selinux ... sendmail 等等不必要的影响并行计算的服务
- 4设置nfs服务 (

```
NFSDIR inode31 /public /public
NFSDIR inode32 /public2 /public2
# NFSDIR node33 /data1 /data1
# NFSDIR inode40 /data2 /data2
BINDHOME /public/home
```

)
 - 如果集群要配置nfs，请按照以下配置，详情查看4.2
 - 配置完成后，在启动集群时，会自动执行，从而正确挂载nfs文件系统。



clusconf 配置操作

5按照预定的流程一键配置集群 (--set-all)

□ 按照如下流程，一键配置集群

- 1) --test-net
- 2) --sys-info
- 3) --set-ssh root
- 4) --sync-file /etc/hosts
- 5) --set-rsh
- 6) --set-service
- 7) --set-nfs
- 8) --sync-user
- 9) --sync-time
- 10) --sync-do "date;df -a" (verify the setting)



clusconf 管理操作

■ 1测试网络 (--test-net)

- 测试/etc/hosts 表中的Ip地址是否都正常，如果不正常，会自动提示

■ 2显示集群信息 (--sys-info)

- 显示标识的节点的操作系统，版本，CPU 型号，核心数，内存容量等各个节点信息

■ 3显示cpu负载(--disp-cpu)

- 显示标识的节点的CPU负载超过5%的进程信息。

■ 4同步集群用户 (--sync-user)

- 同步标识的节点的用户信息，包括用户、组、密码等信息

■ 5添加一个用户到指定的节点(--useradd|--adduser username)

- 按照提示添加用户，包括用户名，家目录，用户组，uid，密码等信息，添加完成后自动同步用户信息，并且提示是否配置该用户的ssh 无密码访问，默认按照提示可配置完成

■ 6删除标识的节点的一个用户(--deluser|--userdel username)

- 删除用户，并自动同步标识的节点的用户信息



clusconf 管理操作

■ 7同步时间 (--sync-time)

- clusconf --sync-time
- 同步标识的节点的时间，并将命令添加到当前管理节点的 /etc/crontab 中，每日0点自动同步集群时间

■ 8同步文件 (--sync-file file|files|path|paths)

- clusconf --sync-file file|files|path|paths
- 可以同步标识的节点的单个文件，多个文件，路径，多个路径

■ 9执行相同的命令 (--sync-do command)

- clusconf --sync-do command
- 在标识的节点同时执行某个命令，如果选择并行模式，则自动对结果进行筛选。

clusconf IPMI相关操作

■ 确认修改<installdir>/etc/ipmi.cfg 正确

■ 1 设置ipmi IP 地址

- 此命令通过s按照ipmi.cfg。

# ipmiaddr	hostname	character	ipmiusername	ipmipasswd
11.11.20.1	node1	comput	root	superuser
11.11.20.2	node2	comput	root	superuser
11.11.20.3	node3	io	admin	admin
11.11.20.4	node4	io	root	superuser
11.11.20.5	node5	admin	root	superuser
11.11.20.6	node6	admin	admin	admin

■ 2 显示ipmi信息

- 通过访问ipmi网络地址，以指定节点的ipmi的设备的ip地址，子网掩码，以及电源状态信息，并对结果进行筛选显示。

■ 3 IPMI开机(--ipmi-poweron)。

- 通过访问ipmi的网络地址，对指定节点使用ipmi控制开机，只要集群上电，即可通过此命令对节点批量开机，从而大大简化系统开机流程和工作量。

■ 4 IPMI关机(--ipmi-poweroff)。

- 通过访问ipmi的网络地址，对指定节点使用ipmi控制挂机，只要集群上电，即可通过此命令对节点批量关机，从而大大简化系统关机流程和工作量。



clusconf IPMI 智能开关机

■ 5 IPMI智能开机(--ipmi-ipoweron)。

- 通过访问ipmi的网络地址，对指定节点使用ipmi控制智能开机。智能开机，指的是根据ipmi.cfg 文件定义的节点角色，按照集群开机的顺序[IO节点，管理节点，计算节点]进行开机，并对开机的结果进行检查。只要集群上电，即可通过此命令对节点批量关机，并且规避了由于开机顺序的依赖关系导致的问题和故障，从而大大简化系统开机流程和工作量。
- 智能开机，需要确认你当前工作节点为管理登陆节点，当前路径为本地路径。

■ 6 IPMI智能关机(--ipmi-ipoweroff)。

- 通过访问ipmi的网络地址，对指定节点使用ipmi控制智能关机，智能关机，指的是根据ipmi.cfg 文件定义的节点角色，按照集群关机的顺序[计算节点，管理节点，IO节点]进行关机，并对关机的结果进行检查。只要集群上电，即可通过此命令对节点批量关机，并且规避了由于关机顺序的依赖关系导致的问题和故障，从而大大简化系统关机流程和工作量。
- 智能关机，需要确认你当前工作节点为管理登陆节点，当前路径为本地路径。



HPC软件环境自动部署工具Clussoft

- HPC软件环境主要包括编译器、常用函数库、并行库、benchmark工具集、各领域应用软件等
- HPC软件环境部署的困难：
 - 绝大部分软件需要编译，容易出错，对实施人员的技术经验要求较高
 - 软件之间的依赖关系复杂
 - 许多软件需要针对不同的硬件平台做不同的配置和优化
 - 手动安装工作量巨大，重复劳动很多，效率低下
- Clussoft的设计功能：
 - 提供HPC软件环境的自动部署
 - 将各个软件的编译安装都脚本化，自动解决软件依赖关系、自动针对软硬件平台选择优化配置
 - 解放劳动力，提供部署效率和质量，将HPC软件实施标准化
 - 适合高性能计算云平台的软件快速部署
 - 模块化设计，开发方便



HPC软件环境自动部署工具Clussoft

■ Clussoft 部分组件：

- clussoft-base——编译器、函数库、并行库等
- clussoft-benchmark——常用benchmark工具（CPU、内存、网络、I/O等）
- clussoft-md——常用开源分子动力学、Monte Carlo等应用软件
- clussoft-qchem——常用开源计算化学、物理、材料类应用软件
- clussoft-bio——生命科学领域常用开源应用软件
- clussoft-gpu——GPGPU开发环境及应用软件（正在开发）
- clussoft-mets——气象海洋类应用软件（正在开发）

■ 总计已支持超过90+个HPC软件



HPC软件环境自动部署工具Clussoft

■ clussoft-base

```
--ofed          Install OFED (InfiniBand drivers and libraries). Please edit "nodelist" before
--intel         Install Intel C/C++/Fortran Compilers and Intel MKL
--openmpi-intel Install OpenMPI built with Intel compiler (with ScaLAPACK)
--openmpi-gnu   Install OpenMPI built with GNU compiler (with ScaLAPACK)
--mvapich2-intel Install MVAPICH2 built with Intel compiler (with ScaLAPACK)
--mvapich2-gnu  Install MVAPICH2 built with GNU compiler (with ScaLAPACK)
--acml          Install ACML (AMD Core Math Library)
--lapack        Install LAPACK library with both GNU and Intel compilers
--atlas         Install ATLAS library with GNU compiler
--plasma        Install PLASMA library with Intel compiler
--plapack       Install PLAPACK library with OpenMPI-Intel
--fftw2         Install FFTW2 library (both float and double precision, with GNU compilers)
--fftw3         Install FFTW3 library (both float and double precision, with GNU compilers)
--gsl           Install GSL (GNU Scientific Library) with GNU compiler
--hdf4          Install HDF4 library with both GNU and Intel compilers
--hdf5          Install HDF5 library with both GNU and Intel compilers
--netcdf        Install netCDF library with both GNU and Intel compilers
--boost         Install Boost C++ library
--petsc         Install PETSc library (with OpenMPI-Intel + MKL/ACML)
```



HPC软件环境自动部署工具Clussoft

■ clussoft-benchmark

```
--hpl-intel  Install HPL Linpack benchmark for Intel Architecture (with OpenMPI-Intel + MKL)
--hpl-amd    Install HPL Linpack benchmark for AMD architecture (with OpenMPI-Intel + ACML)
--hpl        Install HPL Linpack benchmark (auto-select architecture)
--hpcc-intel Install HPCC benchmarks for Intel Architecture (with OpenMPI-Intel + MKL)
--hpcc-amd   Install HPCC benchmarks for AMD Architecture (with OpenMPI-Intel + ACML)
--hpcc       Install HPCC benchmarks (auto-select architecture)
--imb        Install Intel MPI Benchmarks (with OpenMPI-Intel)
--mpigraph   Install mpiGraph benchmark (with OpenMPI-Intel)
--stream     Install Stream memory benchmark (OpenMP threaded with Intel compilers)
--iozone     Install iozone I/O benchmark
--lmbench    Install LMbench benchmarks
--all        Equals to --hpl --hpcc --imb --mpigraph --stream --iozone --lmbench
```



HPC软件环境自动部署工具Clussoft

■ clussoft-qchem

--abinit	Install ABINIT package (with OpenMPI-Intel + MKL/ACML + ScaLAPACK)
--aces3	Install ACES III package (with OpenMPI-Intel + MKL/ACML)
--alps	Install ALPS package (with OpenMPI-Intel + MKL/ACML + FFTW3-double + HDF5)
--ase	Install ASE (Atomic Simulation Environment) Python Environments
--bigdft	Install BigDFT package (with OpenMPI-Intel + MKL/ACML)
--cp2k	Install CP2K package (with OpenMPI-Intel + MKL/ACML+FFTW3-double + ScaLAPACK)
--cpmd	Install CPMD package (with OpenMPI-Intel + MKL/ACML + FFTW2-double)
--dacapo	Install Dacapo package (with OpenMPI-Intel + MKL/ACML + FFTW2-double + NetCDF + Python, need runtime ASE)
--dft++	Install DFT++ package (with OpenMPI-Intel + MKL/ACML + FFTW2-double)
--elk	Install ELK FP-LAPW package (with OpenMPI-Intel + MKL/ACML)
--ergoscf	Install ErgoSCF package (OpenMP threaded with Intel Compiler + MKL/ACML)
--exciting	Install EXCITING FP-LAPW package (with OpenMPI-Intel + MKL/ACML)
--gamess-us	Install GAMESS-US package (with OpenMPI-Intel + MKL/ACML)
--gpaw	Install GPAW package (with OpenMPI-Intel + MKL/ACML + ScaLAPACK + Python, need runtime ASE)
--lmtart	Install LmtART package (serial version with Intel compiler)
--madness	Install MADNESS package (with OpenMPI-Intel + MKL/ACML)
--mpqc	Install MPQC package (with OpenMPI-Intel + MKL/ACML)
--nwchem	Install NWChem package (with OpenMPI-Intel + MKL/ACML)
--octopus	Install Octopus package (with OpenMPI-Intel + MKL/ACML + FFTW3-double + ScaLAPACK + GSL)
--openmx	Install OpenMX package (with OpenMPI-Intel + MKL/ACML + FFTW3-double)
--pwscf	Install PWscf (Quantum-Espresso) package (with OpenMPI-Intel + MKL/ACML + ScaLAPACK + FFTW3-double)
--qbox	Install Qbox package (with OpenMPI-Intel + MKL/ACML + ScaLAPACK + FFTW2-double)
--qmcpack	Install QMCPACK package (with OpenMPI-Intel + MKL/ACML + FFTW3-double + HDF5 + Boost)
--qwalk	Install Qwalk package (with OpenMPI-Intel + MKL/ACML)
--siesta	Install SIESTA package (with OpenMPI-Intel + MKL/ACML + ScaLAPACK)
--yambo	Install Yambo package (with OpenMPI-Intel + MKL/ACML + FFTW3-double)



HPC软件环境自动部署工具Clussoft

■ clussoft-md

```
--ammp      Install AMMP package (serial version only, with Intel Compiler)
--asap      Install Asap package (with OpenMPI-Intel + Python, need runtime ASE)
--ase       Install ASE (Atomic Simulation Environment) Python Environments
--dl_class  Install DL_POLY Classic package (with OpenMPI-Intel)
--dl_monte  Install DL_MONTE Monte Carlo package (with OpenMPI-Intel)
--dl_poly   Install DL_POLY 4 package (with OpenMPI-Intel)
--ego       Install EGO package (with OpenMPI-Intel)
--espresso  Install ESPResSo package (with OpenMPI-Intel + FFTW3-double)
--gromacs   Install GROMACS package (both single and double precision, with OpenMPI-Intel + MKL/ACML+FFTW3)
--lammps    Install LAMMPS package (with OpenMPI-Intel + FFTW2-float)
--mdynamix  Install MDynaMix package (with OpenMPI-Intel)
--moldy     Install Moldy package (with OpenMPI-Intel)
--mpdyn     Install MPDyn package (with OpenMPI-Intel)
--namd      Install NAMD package (with OpenMPI-Intel + FFTW3-float)
--openmd    Install OpenMD package (with OpenMPI-Intel + FFTW3-double)
--tinker    Install Tinker package (OpenMP threaded with Intel Compiler + FFTW3-double)
--towhee    Install Towhee Monte Carlo package (with OpenMPI-Intel)
--xmd       Install XMD package (POSIX threaded with Intel compiler)
```



HPC软件环境自动部署工具Clussoft

■ clussoft-bio

```
--abyss          Install ABySS sequence assembler package (with OpenMPI-Intel + Boost)
--allpathslg     Install ALLPATHS-LG sequence assembler package (OpenMP threaded with GNU compiler)
--amos           Install AMOS sequence assembler package (serial version, Intel compiler + Boost)
--autodock       Install AutoDock moleculer docking package (serial version with Intel compiler)
--autodock-vina  Install AutoDock Vina moleculer docking package (serial version with Intel compiler)
--blast          Install NCBI-BLAST+ sequence searching package (POSIX threaded with GNU compiler)
--clustal-omega  Install Clustal Omega sequence alignment package (OpenMP threaded with Intel compiler)
--clustalw       Install ClustalW sequence alignment package (serial version with Intel compiler)
--clustalw-mpi   Install ClustalW-MPI sequence alignment package (with OpenMPI-Intel)
--dock           Install Dock moleculer docking package (with OpenMPI-GNU)
--exonerate      Install EXONERATE sequence alignment package (POSIX threaded with GNU compiler)
--fasta          Install FASTA sequence searching package (MPI version with OpenMPI-Intel)
--fsa            Install FSA sequence alignment package (serial with GNU compiler)
--hmmer          Install HMMER sequence searching package (both OpenMPI-Intel and POSIX threaded version)
--mira           Install MIRA sequence assembler package (serial version, Intel compiler + Boost)
--mpiblast       Install mpiBLAST sequence searching package (with OpenMPI-Intel)
--mpihmmer       Install mpiHMMER sequence searching package (with OpenMPI-Intel)
--mummer         Install MUMmer sequence alignment package (serial version with Intel compiler)
--velvet         Install Velvet sequence assembler package (OpenMP threaded with Intel compiler)
--wgs            Install WGS sequence assembler package (with GNU compiler)
```

