Linux高性能计算集群配置

李会民
hmli@ustc.edu.cn

中国科学技术大学 超级计算中心

2014-07-18

内容

高性能计算、超级计算、并行计算

- 高性能计算(HPC) 指通常使用很多处理器（作为单个机器的一部分）或者某一集群中组织的几台计算机（作为单个计算资源操作）的计算系统和环境。有许多类型的HPC系统，其范围从标准计算机的大型集群，到高度专用的硬件。大多数基于集群的HPC系统使用高性能网络互连，比如那些来自InfiniBand或Myrinet的网络互连。基本的网络拓扑和组织可以使用一个简单的总线拓扑，在性能很高的环境中，网状网络系统在主机之间提供较短的潜伏期，所以可改善总体网络性能和传输速率。
- 一般不区分高性能计算、超级计算、并行计算之间的差别。

(a) 天津超算：天河-1A

(b) 深圳超算：曙光星云

(c) 济南超算：神威蓝光

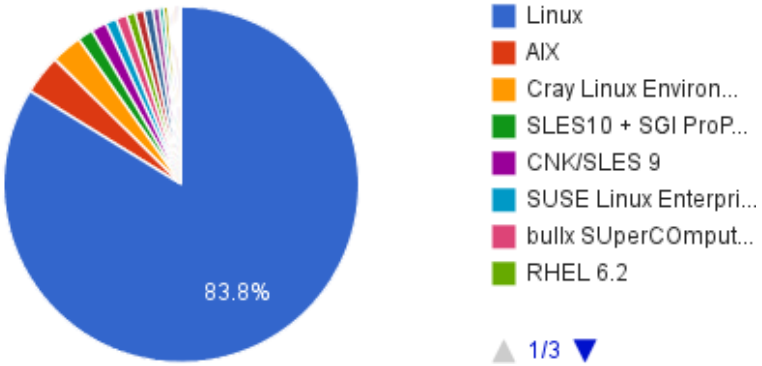
影响高性能计算的主要因素

- 硬件:
  - CPU:
    - 主要参数: 主频、核数、并发数、Cache
    - 评测程序: SPEC、HPL(High Performance Linpack)
  - 内存:
    - 主要参数: 大小、主频、CL延迟
    - 评测程序: Stream
  - IO能力:
    - 主要参数: 缓存、转速、接口速率
    - 评测程序: IOZone、dd
  - 网络:
    - 主要参数: 带宽、延迟
    - 评测程序: IMB(Intel MPI Benchmark)
  - 能耗:
- 软件:
  - 编译器、数值函数库、并行库
- 设置:
  - 硬件
  - 操作系统
  - 软件

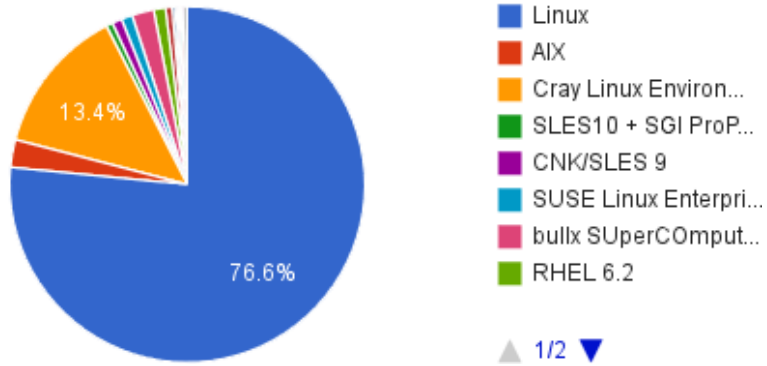
2012年11月的TOP500中Linux份额

高性能计算系统排名：http://www.top500.org

Operating System System Share



Operating System Performance Share



Operating System	Count	System Share (%)	Rmax (GFlops)	Rpeak (GFlops)	Cores
Linux	419	83.8	124122700	177021632	12328716
AIX	18	3.6	4072666	5099712	182976
Cray Linux Environment	14	2.8	21742588	32301256	1034656
SLES10 + SGI ProPack 5	7	1.4	960800	1096704	94208
CNK/SLES 9	7	1.4	1453422	1749811	528384
SUSE Linux Enterprise Server 11	5	1	1624382	1921199	94752
bullx SUperCOmputer Suite A.E.2.1	5	1	3241378	3961958	183424
RHEL 6.2	4	0.8	1738900	2132582	102528
CentOS	4	0.8	955100	1182927	88928
CNL	4	0.8	453460	587565	60144
Redhat Linux	3	0.6	311080	384785	42144
Windows HPC 2008	2	0.4	314300	460398	38028
RedHat Enterprise 5	2	0.4	177740	200271	17088
Windows Azure	1	0.2	151300	167731	8064
Super-UX	1	0.2	122400	131072	1280
SUSE Linux	1	0.2	274800	308283	26304
RHEL 6.1	1	0.2	230600	340915	37056
Open Solaris	1	0.2	110600	121282	12032
Cell OS	1	0.2	81171	105830	5088

Unix/Linux操作系统发行版本

一些主要发行版：

- Linux：
  - 常见： Android, Arch, CentOS, Debian, Fedora, Gentoo, Mandriva, Red Hat Enterprise Linux(RHEL), Slackware, SUSE Linux Enterprise Desktop(SLED), SUSE Linux Enterprise Server(SLES), OpenSuSE, Ubuntu, . . .
  - 通用高性能计算系统常见： RHEL系（RHEL、CentOS、Scientific Linux-SL）、SUSE系
- Unix：
  - 学院派BSD： FreeBSD, OpenBSD, NetBSD, . . .
  - 商业Unix： IBM AIX, HP UX, Sun Solaris, OpenSolaris[1], Mac OS X[2], iOS, SGI IRIX, . . .

Linux、BSD发布版： http://distrowatch.com/

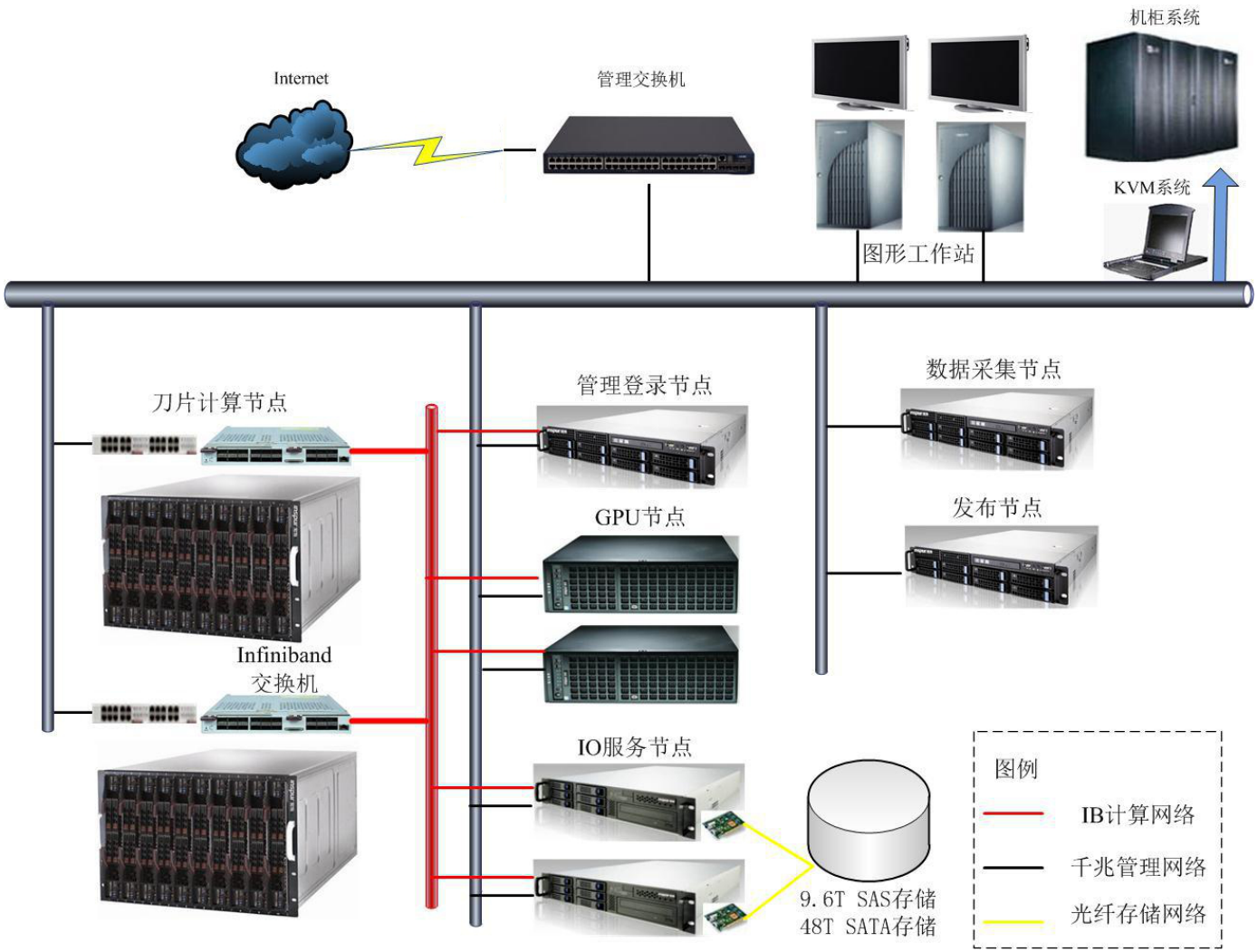
[1] Sun公司按照CDDL授权开源

[2] 以FreeBSD源代码和Mach微内核为基础

CentOS系统

- CentOS Linux基于Red Hat Enterprise Linux(RHEL)
- 本文除非特别说明，否则都以CentOS 6 x86_64 Linux系统为例
- CentOS yum源目录：*/etc/yum.repos.d*
- CentOS常用命令：
  - 安装软件包：*yum␣install␣packagename*
  - 搜索软件包：*yum␣search␣string*
  - 查看软件包简要信息：*yum␣list␣packagename*
  - 查看软件包详细信息：*yum␣info␣packagename*
  - 列出已经安装软件包：*yum␣list␣installed*
  - 查看服务状态：*chkconfig␣--list␣servicename*
  - 增加服务：*chkconfig␣--add␣servicename*
  - 删除服务：*chkconfig␣--del␣servicename*
  - 设置系统启动时服务启动：*chkconfig␣servicename␣on*
  - 设置系统启动时服务关闭：*chkconfig␣servicename␣off*

集群规划 I


Internet
管理交换机
机柜系统
KVM系统
图形工作站
刀片计算节点
管理登录节点
数据采集节点
GPU节点
发布节点
Infiniband
交换机
IO服务节点
9.6T SAS存储
48T SATA存储
图例
IB计算网络
千兆管理网络
光纤存储网络

集群规划 II

- 主节点（管理、IO、用户登录，服务端）：
  - 节点名： admin
  - 内网eth0网卡IP： 192.168.1.254
  - 编辑*/etc/hosts*，设定节点名与IP对应：

```
127.0.0.1       localhost .localdomain      localhost
192.168.1.254  admin
192.168.1.1    node1
192.168.1.2    node2
```

- 计算节点（客户端）：
  - 节点名： node1、node2、...
  - 内网eth0网卡IP： 192.168.1.1、192.168.1.2、...
  - 编辑*/etc/hosts*，添加如下内容，以便能根据节点名找到主节点IP：

```
192.168.1.254  admin
```

NFS：网络文件系统

- NFS(Network File System)：网络文件系统
- 各计算节点需共享文件，比如*/home*和*/opt*等

NFS服务端设置

- 安装所需的NFS包：*yum␣-y␣install␣nfs-utils*
- 设置NFS服务系统启动时自启动：*chkconfig␣nfs␣on*
- 重启NFS服务：*service␣nfs␣restart*
- 编辑*/etc/exports*，添加NFS共享目录及允许IP[3]：

```
/home 192.168.1.0/24(rw,async,no_root_squash)
/opt  192.168.1.0/24( rw,async,no_root_squash)
```

- 刷新NFS设置：*exportfs␣-ra*
- 查看NFS状态：*exportfs␣-v*

```
/home  192.168.1.0/24(rw,async,wdelay,no_root_squash,no_subtree_check)
/opt   192.168.1.0/24(rw,async,wdelay,no_root_squash,no_subtree_check)
```

[3]同步sync保证正确但降低性能，一般用异步async

NFS客户端设置

- 编辑*/etc/fstab*，添加NFS共享目录[4]：

```
admin:/home /home nfs  defaults , nfsvers=3 0  0
admin:/opt  /opt  nfs  defaults , nfsvers=3 0  0
```

- 挂载文件系统：*mount␣−a*
- 查看挂载情况：*mount*

```
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/sda1 on /boot type ext3 (rw)
/dev/sda5 on /tmp type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
admin:/home on /home type nfs (rw,nfsvers=3,addr=192.168.1.254)
admin:/opt on /opt type nfs (rw,nfsvers=3,addr=192.168.1.254)
```

[4]NFS V4版本默认采用tcp协议，导致性能降低，建议采用V3版本

NIS：网络信息服务

NIS(Network Information Service)：网络信息服务

- 对主机帐号等系统信息提供集中管理的网络服务
- 用户登录任何一台NIS客户机都会从NIS服务器进行登录认证，可实现用户帐号的集中管理
- 主要同步信息：用户信息、节点名信息等
- 在NIS环境中，有三种类型的主机：
  - 服务器(master)：充当主机配置信息的中央数据库，保存着用户帐号、组帐号等配置信息的权威副本
  - 从服务器(slave)：保存这些信息的冗余副本
  - 客户机(client)：使用这些信息

NIS服务端设置 I

- 安装所需包：*yum␣-y␣ install ␣ypserv␣ypbind␣yp-tools␣rpcbind*
- 设定NIS域名：
  - 在*/etc/sysconfig/network*中添加域名信息：

```
NISDOMAIN=mydomain.edu
```

  - 设置系统启动时自动设置域名，在*/etc/rc.local*中添加：

```
/bin/nisdomainname mydomain.edu
```

  - 使当前域名生效：*nisdomainname␣mydomain.edu*
  - 修改*/var/yp/securenets*，设置允许客户端IP范围：

```
host   127.0.0.1
255.255.255.0   192.168.1.0
```

NIS服务端设置 II

- 修改*/etc/yp.conf*[5]：

```
ypserver  127.0.0.1
```

- 修改需要同步信息的配置文件*/var/yp/Makefile*，只同步用户信息、组信息和节点名信息：

```
all:  passwd group hosts #rpc service netid protocols mail \
# netgrp shadow publickey networks ethers bootparams printcap \
# amd.home auto.master auto.home auto.local passwd.adjunct \
# timezone locale netmasks
```

- 初始化：*/usr/lib64/yp/ypinit␣−m*
- 启动服务：
  - rpc守护进程：*service␣rpcbind␣start*
  - 服务守护进程：*service␣ypserv␣start*
  - 客户守护进程：*service␣ypbind␣start*

NIS服务端设置 III

- 用户运行*yppasswd*修改密码守护进程：*service␣yppasswdd␣start*
- 设置系统启动时自启动：
  - rpc守护进程：*chkconfig␣rpcbind␣on*
  - 服务守护进程：*chkconfig␣ypserv␣on*
  - 客户守护进程：*chkconfig␣ypbind␣on*
  - 用户运行*yppasswd*修改密码守护进程：*chkconfig␣yppasswdd␣on*
- 设置定时推送用户等信息：
  - 运行*crontab␣−e*，输入内容：

```
# m h  dom mon dow  command
*/5 * * * * cd /var/yp; make >/dev/null
```

[5]服务端也是自己的客户端

NIS客户端设置 I

- 安装所需包：*yum␣-y␣ install ␣ypbind␣yp-tools␣rpcbind*
- 保证客户端可通过服务端名字获得对应IP，修改*/etc/hosts*：

```
192.168.1.254  admin
```

- 设定NIS域名：
  - 修改*/etc/sysconfig/network*：

```
NISDOMAIN=mydomain.edu
```

  - 设置系统启动时自动设置域名，在*/etc/rc.local*中添加：

```
/bin/nisdomainname mydomain.edu
```

- 设定服务节点，修改*/etc/yp.conf*：

NIS客户端设置 II

```
domain mydomain.edu server admin
```

- 使当前域名生效：*nisdomainname␣admin*
- 设置所需要同步的信息，修改*/etc/nsswitch.conf*

```
passwd:     files   nis
shadow:     files   nis
group:      files   nis
hosts :     files   nis  dns
```

- 启动服务：*service␣ypbind␣start*
- 启动rpc守护进程：*service␣rpcbind␣start*
- 设置系统启动时自启动rpc守护进程：*chkconfig␣rpcbind␣on*

NIS客户端设置 III

- 设置系统启动时自启动： *chkconfig␣ypbind␣on*
- 测试：
  - 检查是否启动： *ypwhich*
  - 测试用户hmli信息： *id␣hmli*
  - 检查同步的文件： *yptest*

quota：磁盘配额

- 磁盘配额就是管理员可以为用户所能使用的磁盘空间进行配额限制，每一用户只能使用最大配额范围内的磁盘空间。
- 设置磁盘配额后，可以对每一个用户的磁盘使用情况进行跟踪和控制，通过监测可以标识出超过配额报警阈值和配额限制的用户，从而采取相应的措施。
- 磁盘配额管理功能的提供，使得管理员可以方便合理地为用户分配存储资源，可以限制指定账户能够使用的磁盘空间，这样可以避免因某个用户的过度使用磁盘空间造成其他用户无法正常工作甚至影响系统运行避免由于磁盘空间使用的失控可能造成的系统崩溃，提高了系统的安全性。

磁盘配额quota设置

- 安装所需包：*yum −y install quota*
- 修改*/etc/fstab*[6]，增加 usrjquota=aquota.user,grpjquota=aquota.group,jqfmt=vfsv0：

```
/dev/sda5  /home ext3    defaults , usrjquota=aquota.user , grpjquota=aquota.group,jqfmt=vfsv0 1 2
```

- 重新挂载：*mount −o remount /home*
- 检查并生成所需信息[7]：*quotacheck −cvug /home*
- 开启配额：*quotaon −vug /home*
- 设置hmli用户配额：*edquota hmli*

```
Disk quotas for user hmli (uid 502):
  Filesystem                   blocks       soft       hard     inodes     soft     hard
  /dev/sda5                  13907432   49000000   50000000      23491        0        0
```

格式说明：

文件系统 当前占用块大小 块大小软限制 块大小硬限制 当前inodes大小 inodes软限制 inodes硬限制

- 关闭磁盘限额：*quotaoff −vug /home*
- 查看hmli用户磁盘配额：*quota hmli*

[6]假设/dev/sda5为需要设置的分区

[7]启动quota前必须做

网络批量系统安装

大规模的部署Red Hat Linux系（CentOS等）操作系统

- 避免手工安装的繁琐
- 避免出错，保证一致性

网络批量系统安装

大规模的部署Red Hat Linux系（CentOS等）操作系统

- 避免手工安装的繁琐
- 避免出错，保证一致性
- dd硬盘或nc网络对拷对拷：
  - 1->2->4->8->...
  - 需要拔插硬盘或光盘或网络启动、挂载分区、修改IP地址、节点名等，繁琐

网络批量系统安装

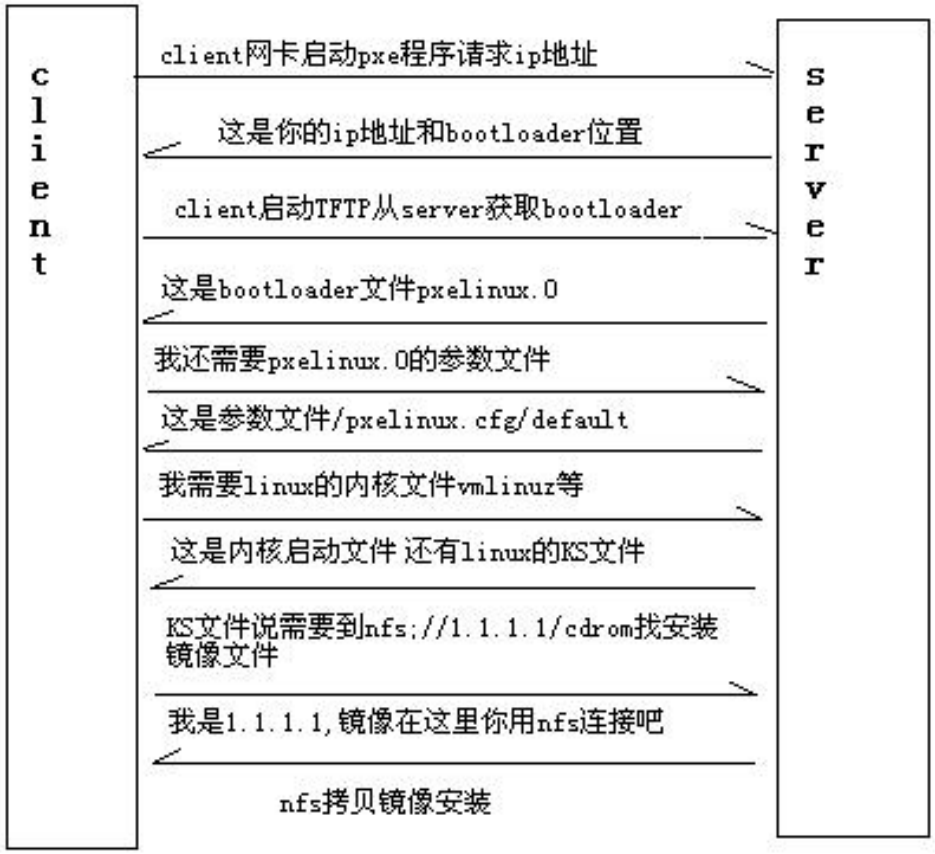
大规模的部署Red Hat Linux系（CentOS等）操作系统

- 避免手工安装的繁琐
- 避免出错，保证一致性
- dd硬盘或nc网络对拷对拷：
  - 1->2->4->8->...
  - 需要拔插硬盘或光盘或网络启动、挂载分区、修改IP地址、节点名等，繁琐
- kickstart网络安装：
  - 许多系统管理员宁愿使用自动化的安装方法来安装Red Hat Linux系
  - 为满足这种需要，Red Hat创建了kickstart安装
  - 使用kickstart，系统管理员可创建一个文件，这个文件包含了在典型的安装过程中所遇到问题的答案
  - kickstart文件可以存放于单一的服务器上，在安装过程中被独立的机器所读取
  - 此安装方法可以支持使用单一kickstart文件在多台机器上安装Red Hat Linux，对于网络和系统管理员来说是个理想的选择
  - kickstart给用户提供了一种自动化安装装Red Hat Linux的方法
  - CentOS等Red Hat系Linux发行版也支持kickstart安装

PXE网络安装 I

- PXE(Pre-boot Execution Environment)是由Intel设计的协议，它可以使计算机通过网络启动
- 协议分为client和server两端，PXE client在网卡的ROM中，当计算机引导时，BIOS把PXE client调入内存执行，并显示出命令菜单，经用户选择后，PXE client将放置在远端的操作系统通过网络下载到本地运行
- 在其启动过程中，客户端请求服务器分配IP地址，之后PXE Client使用TFTP Client通过TFTP(Trivial File Transfer Protocol)协议下载启动安装程序所需的文件
- PXE网络安装：客户机通过支持PXE的网卡向网络中发送请求DHCP信息的广播请求IP地址等信息，DHCP服务器给客户端提供IP地址和其它信息（TFTP服务器、启动文件等），之后请求并下载安装需要的文件

PXE网络安装 II


client
server
client网卡启动pxe程序请求ip地址
这是你的ip地址和bootloader位置
client启动TFTP从server获取bootloader
这是bootloader文件pxelinux.0
我还需要pxelinux.0的参数文件
这是参数文件/pxelinux.cfg/default
我需要linux的内核文件vmlinuz等
这是内核启动文件 还有linux的KS文件
KS文件说需要到nfs://1.1.1.1/cdrom找安装镜像文件
我是1.1.1.1,镜像在这里你用nfs连接吧
nfs拷贝镜像安装

DHCP安装和配置 I

- 安装dhcp包：*yum␣−y␣ install ␣dhcp*
- 复制模板文件：*cp␣/usr/share/doc/dhcp−4.1.1/dhcpd.conf.sample␣/etc/dhcp/dhcpd.conf*
- 修改*/etc/dhcp/dhcpd.conf*，设置PXE文件及IP与MAC地址的对应：

```
# option definitions common to all supported networks ...
option domain−name ”mydomain.edu”; # 域名
option domain−name−servers ns1.ustc.edu.cn; # 域名服务器

default −lease−time 600;
max−lease−time 7200;

subnet 192.168.1.0 netmask 255.255.255.0 {
    option routers                  192.168.1.254; # 路由网关
    option subnet−mask              255.255.255.0;
    option nis−domain               ”ustc .edu.cn”; # 域名
    option domain−name              ”ustc .edu.cn”;
    option domain−name−servers 192.168.1.254; #域名服务器
    filename ”/pxelinux .0”; #PXE 文件

    option time− offset                 −18000; # Eastern Standard Time
    range dynamic−bootp 192.168.1.1 192.168.1.243; # bootp IP 范围
    default −lease−time 21600;
    max−lease−time 43200;
```

DHCP安装和配置 II

```
    host node1 {
        hardware ethernet 40:61:86:ED:95:30; # IP 地址与 MAC 地址对应
        fixed-address 192.168.1.1;
    }
    host node2 {
        hardware ethernet 40:61:86:ed:93:7e;
        fixed-address 192.168.1.2;
    }
}
option space PXE;
class "pxeclients" {
    match if substring(option vendor-class- identifier , 0, 9) = "PXEClient";
    next-server 192.168.1.254; # PXE 服务器
    filename "/pxelinux.0"; # PXE 文件
}
```

- 如果不知道MAC地址，那么将会自动分配随机地址，客户端系统装好后可以修改客户端配置设置成固定IP
- 启动DHCP服务： *service␣dhcp␣start*
- 设置系统启动时自启动服务： *chkconfig␣dhcp␣on*

配置TFTP服务器

PXE安装时，客户机使用TFTP协议从服务器下载引导文件并执行

- 安装配置TFTP服务器：*yum␣-i␣ install ␣ tftp -server*
- tftp服务由xinetd服务管理，编辑 */etc/xinetd.d/tftp*：

```
# default: off
# description: The tftp server serves files using the trivial file transfer \
#       protocol. The tftp protocol is often used to boot diskless \
#       workstations, download configuration files to network-aware printers, \
#       and to start the installation process for some operating systems.
service tftp
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /tftpboot # 目录名
        disable                 = no # 改为：no
        per_source              = 11
        cps                     = 100 2
        flags                   = IPv4
}
```

- 重启服务：*service ␣ xinetd ␣ restart*

PXE引导配置 I

- PXE启动映像文件由syslinux软件包提供，CentOS光盘中已提供，以下以ISO镜像中为例
- 安装syslinux以获取*pxelinux.0*等：*yum␣-y␣install␣syslinux*
- 将*pxelinux.0*复制到*/tftpboot*：
  cp␣/usr/share/syslinux/pxelinux.0␣/tftpboot
- 挂载第一张DVD镜像：
  mount␣-o␣loop␣CentOS-6.2-x86_64-bin-DVD1.iso␣/mnt
- 将安装光盘目录中的启动文件复制*/tftpboot*
  cp␣/mnt/images/pxeboot/{vmlinuz,initrd.img}␣/tftpboot
- 创建存放客户端的配置文件*default*：
  cp␣/mnt/isolinux/isolinux.cfg␣/tftpboot/pxelinux.cfg/default

PXE引导配置 II

- 修改配置*/tftpboot/pxelinux.cfg/default*:

```
default linux
#prompt 1 # 不要提示，直接进行安装
timeout 60 # 提示时的等待时间

display boot.msg

menu background splash.jpg
menu title Welcome to CentOS 6.2!

label linux
  menu label ^ Install or upgrade an existing system
  menu default
  kernel vmlinuz
  append initrd = initrd .img
  append ksdevice=eth0 load_ramdisk=1 initrd = initrd .img network \
      ks=nfs :192.168.1.254:/ tftpboot /ks.cfg noipv6 devfs=nomount selinux=0 nostorage \
     driverload =sd_mod:mptbase:mptscsih:mptsas:aacraid :mptscsih: libata :megaraid_sas:scsi_mod
# 主要为上面三行，设置 ks 文件，内核引导参数等
label local
  menu label Boot from ^local drive
  localboot 0
```

配置NFS目录

将*/mnt*通过NFS共享给客户端:

- 编辑*/etc/exports*，增加下面一行:

```
/mnt  192.168.1.0/24(rw,async,no_root_squash)
```

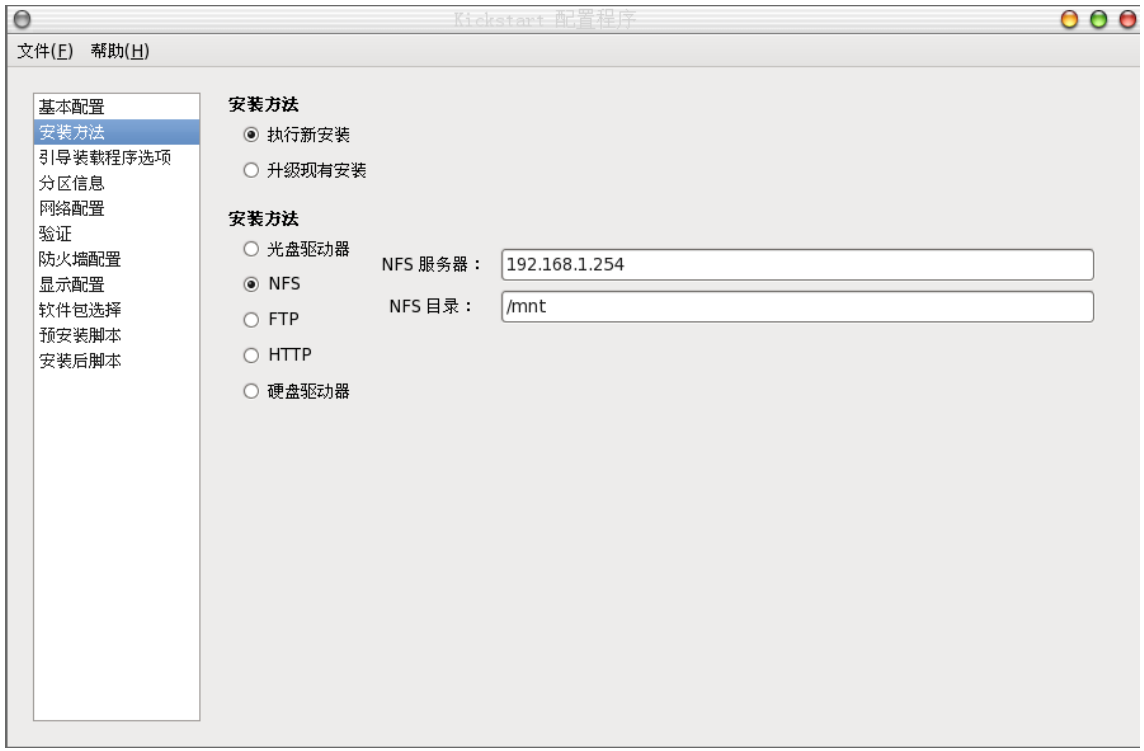
- 刷新NFS配置: *exportfs␣-ra*

配置Kickstart安装 I

- 通常，在安装操作系统的过程需大量的人机交互过程，减少交互过程，为提高安装效率Red Hat Linux开始支持称为kickstart的功能
- 只需事先定义好一个kickstart自动应答配置文件（通常存放在安装服务器上），并让安装程序知道该配置文件的位置，在安装过程中安装程序就可以自己从该文件中读取安装配置，这样就避免了繁琐的人机交互，实现无人值守的自动化安装
- 安装好一台CentOS机器，安装程序都会创建一个kickstart配置文件*/root/anaconda-ks.cfg*，记录真实安装配置

配置Kickstart安装 II

- 使用*system−config− kickstart* 命令配置*ks.cfg*文件：
  - 安装所需包：*yum␣−i␣ install ␣system−config− kickstart*
  - 运行图形界面进行设置：*system−config− kickstart*
  - 载入*/root/anaconda−ks.cfg*作为模板：文件->打开文件，选择*/root/anaconda−ks.cfg*
  - 在此界面上进行配置并保存为*/tftpboot/ks.cfg*



配置Kickstart安装 III

- 修改*/tftpboot/ks.cfg*，增减所需安装的软件包等：

```
#platform=x86, AMD64, 或 Intel EM64T
#version=DEVEL
# Firewall configuration
firewall --disabled
# Install OS instead of upgrade
install
# Use NFS installation media
nfs --server=192.168.1.254 --dir=/mnt # 设置所需要下载文件的协议及服务端IP和目录
# Root password
rootpw --iscrypted $6$fQW/BV8Uw/af07Vh$YnA8l5jZtVSvAOEKMkzpriC6pl/ntXnrRK1cPZgsZSS0qv6v6sGjdxR.qc8zC15
# System authorization information
auth --useshadow --passalgo=sha512 --enablenis --nisdomain=mydomain.edu --nisserver=admin
# Use text mode install
text
# System keyboard
keyboard us
# System language
lang en_US
# SELinux configuration
selinux --disabled
# Do not configure the X Window System
skipx
# Installation logging level
```

配置Kickstart安装 IV

```
logging --level=info

# System timezone
timezone  Asia/Shanghai
# Network information
network  --bootproto=dhcp --device=eth0 --onboot=on
# System bootloader  configuration
bootloader --location=mbr
# Clear the Master Boot Record
zerombr
# Partition  clearing  information
clearpart  --all -- initlabel
# Disk  partitioning  information  # 设置硬盘分区
part  /boot --fstype="ext3" --size=200
part  /  --fstype="ext4" --size=10000
part  swap --fstype="swap" --size=6000
part  /tmp --fstype="ext4" --grow --size=1000

%packages # 增减所需要安装的软件包
@base
@network-server
@performance
@storage-server
@system-admin-tools
cpufrequtils
```

配置Kickstart安装 V

```
dhcp
sdparm
yp-tools
 tree
tuned
tuned- utils
vim-enhanced
ypbind
-lvm2
-nano
-pcmciautils
-plymouth
- rfkill
-rsync
-system-config-firewall-tui
-system-config-network-tui
-unzip
-vconfig
-wireless-tools
-vim-minimal
%end
```

客户端网络启动安装

- 设置客户端BIOS选择从网卡启动，具体方法因BIOS版本不同而异
- 网卡中的PXE代码会联系DHCP服务器来获取IP地址以及启动镜像，然后启动镜像被载入并运行
- 安装完成后，安装程序会提示你重新启动机器。重新启动机器时切记要在BIOS里改成从硬盘启动。如仍然从光盘启动机器，又会重复前面的自动安装步骤

ssh免输密码访问

- MPI并行程序运行时，需要无需输入密码，因此需要配置ssh或rsh免输密码访问[8]
- ssh可采用：
  - 基于密钥：适合用户自己设置
  - 基于主机：适合系统管理员设置，无需用户自己设置

[8]建议ssh，rsh一般不再使用

ssh免输密码访问：基于密钥

用户主目录*/home*是共享时，用户使用自己的账户运行以下命令：

- 生成密钥：*ssh−keygen*
- 追加到*~/.ssh/authorized_keys*：
  cat␣~/.ssh/id_rsa.pub␣>>~/.ssh/authorized_keys
- 设置权限，禁止他人可读写：
  - *chmod␣700␣~/.ssh*
  - *chmod␣600␣~/.ssh/authorized_keys*
- 以下与集群设置无关，只是告知一个方法方便访问linux系统：
  - 可以设置在node1节点上远程免输密码访问node2节点：
  - 在node1上：*ssh−copy−id␣username@node2*

ssh免输密码访问：基于主机 I

采用root账户设置，设置后其他用户无需自己设置即可免输密码访问

- *ssh−keyscan*获取各节点rsa，生成*/etc/ssh/ssh_known_hosts*：

```
#!/bin/sh
NODES='admin'
for i in seq 1 100 # 生成 100 个节点的，注意： 为键盘左上角的反引号 ， 不是单引号 '
do
    NODES=$NODES"␣"node$i
done
ssh−keyscan −t rsa $NODES | sort | \
sed −e 's/^node\([[:digit:]]*\)/node\1,192.168.1.\1/' >>/etc/ssh/ssh_known_hosts
```

- 生成*/etc/ssh/shosts.equiv*：

```
#!/bin/sh
echo admin >>/etc/ssh/shosts.equiv
for i in seq 1 100
do
    echo node$i >>/etc/ssh/shosts.equiv
done
```

ssh免输密码访问：基于主机 II

- 编辑*/etc/ssh/sshd_config*:

```
HostbasedAuthentication  yes
IgnoreRhosts  no
```

- 编辑*/etc/ssh/ssh_config*:

```
Host *
    HostbasedAuthentication  yes
    EnableSSHKeysign yes
```

- 可在主节点上设置，设置好后复制这四个文件到其它节点*/etc/ssh*
- 各节点重启ssh服务：*service ␣sshd␣ restart*

NTP：时间服务

- Network Time Protocol(NTP)：网络时间协议
- 目的：同步各节点时见，保持各节点的时间一致性

NTP服务端设置

服务端设置:

- 安装NTP服务: *yum␣-y␣install␣ntp*
- 编辑*/etc/ntp.conf*:

```
# Hosts on local network are less restricted .
 restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap # 设置客户端范围
# Use public servers from the pool.ntp.org project .
# Please consider joining the pool (http :// www.pool.ntp.org/join.html).
server time.ustc.edu.cn # 科大时间服务器
server 0.rhel.pool.ntp.org
server 1.rhel.pool.ntp.org
server 2.rhel.pool.ntp.org
```

- 设置系统启动时启动: *chkconfig␣ntpd␣on*
- 重启服务: *servers␣ntpd␣restart*
- 查看服务状态: *ntpq␣-p*

NTP客户端设置

- 安装所需包：*yum␣-y␣install␣ntpdate*
- 客户端设置整点对时：*crontab␣-e*

```
0 * * * *   /usr/sbin/ntpdate  admin
```

- 服务端重启ntpd服务后，在一定时间内客户端运行*ntpdate␣admin*时出现下述信息，请等待一会儿再测试：

```
18 Nov 14:59:29 ntpdate[10863]: no server suitable for synchronization found.
```

- 查看服务状态：*ntpq␣-p␣admin*

```
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*netfee.ustc.edu 195.113.144.201  2 u   16   64   77    0.233   -4.352   1.593
+dns2.synet.edu. 118.143.17.82    2 u    9   64   77   38.886   -2.280   0.947
+dns1.synet.edu. 118.143.17.82    2 u   13   64   77   35.268   -3.307   0.536
-2001:da8:9000:: 118.143.17.82    2 u    7   64   77  158.884  -17.207   4.399
```

内网客户端访问外网：iptables转发

客户端通过服务端连接外网进行升级及软件安装等

- 服务端，假设eth1为外网网卡，eth0为内网网卡，运行下述脚本：

```
#!/bin/sh
echo 1 > /proc/sys/net/ipv4/ip_forward # 打开转发

modprobe ip_conntrack_ftp # 本机做 FTP 时需用
modprobe ip_nat_ft  # 通过本机的 FTP 需用
iptables -t nat -A POSTROUTING -j SNAT -o eth1 --to eth1-IP # 注意为 eth1 及 eth1 IP
```

- 客户端只要设置为默认走与服务端的内网IP即可，类似：
  ip␣route␣add␣default␣via␣192.168.1.254

集群批量设置

多台机器如何管理?

集群批量设置

多台机器如何管理?

- 一台台登录上去：繁琐、容易出错、速度慢

集群批量设置

多台机器如何管理?

- 一台台登录上去：繁琐、容易出错、速度慢
- 利用for循环、Expect脚本等处理：需要自己编写

集群批量设置

多台机器如何管理?

- 一台台登录上去：繁琐、容易出错、速度慢
- 利用for循环、Expect脚本等处理：需要自己编写
- 集群管理软件：省时省力

集群批量设置

多台机器如何管理?

- 一台台登录上去：繁琐、容易出错、速度慢
- 利用for循环、Expect脚本等处理：需要自己编写
- 集群管理软件：省时省力
- 常见集群管理软件：pdsh、mussh、synctool、xcat、clusterssh、The Cluster Command and Control(C3)

pdsh：并行SHELL

- pdsh：http://sourceforge.net/projects/pdsh/
- CentOS官方源不含pdsh，需添加第三方epel源，参见：http://lug.ustc.edu.cn/wiki/mirrors/help/epel
- 所有节点都安装：*yum␣-y␣ install ␣pdsh*
- 用法：
  - 并行执行命令：*pdsh␣-w␣node[1-100]␣-x␣node[81-90]␣hostname*
  - 并行复制到远端：
    pdcp␣-w␣node1,node[12-18]␣intel.sh␣/etc/profile.d/
  - 并行从远端复制到本地，本地名自动加.节点名区分：
    rpdcp␣-w␣node1,node12␣/etc/profile.d/intel.sh␣/tmp

Expect：交互式脚本语言

- 管理员利器，交互式脚本语言：Expect
- 基于Tcl语言

```
#!/usr/bin/expect
for { set i 1 } { $i < 101 } { incr i} {
    set node node$i
    spawn scp –r .ssh $node:
    expect ”connecting”; send ”yes\r”
    expect ”password”; send ”yourpassword\r”
    expect ”root”; puts ”over”
}
```

编译环境

- C/C++、FORTRAN编译器（支持OpenMP）：GCC、Intel、PGI、NAG
- MPI环境： Open MPI、MPICH、MPICH2、MVAPICH、MVAPICH2、Intel MPI、HP MPI、LAM MPI

C/C++/FORTRAN编译环境

- GCC：*yum␣−y␣ install ␣gcc␣gcc−c++␣gcc−gfortran*
- Intel：
  - 下载编译器压缩包，解压缩后到解压缩后类似目录*l_ccompxe_2011.9.293*下执行*./ install .sh*
  - 当前登录设置环境变量：*.␣/opt/ intel /bin/compilervars .sh␣ intel64*
  - 设置默认登录后的环境变量，编辑*/etc/profile.d/intel.sh*：

```
. /opt/intel/bin/compilervars.sh intel64
```

Open MPI编译环境

- Open MPI安装：
  - 下载：http://www.open-mpi.org/software/ompi/v1.8/downloads/openmpi-1.8.1.tar.bz2
  - 解压缩：*tar␣xvf␣openmpi-1.8.1.tar.bz2*
  - 配置：
    FC=ifort␣CC=icc␣CXX=icpc␣./configure␣--prefix=/opt/openmpi-1.8.1
  - 编译：*make*
  - 安装：*make␣install*
- 配置环境变量，编辑*/etc/profile.d/openmpi.sh*：

```
export OPENMPI=/opt/openmpi-1.8.1
export PATH=$OPENMPI/bin:$PATH
export MANPATH=$MANPATH:$OPENMPI/share/man
```

作业调度系统

- 在一个大型系统内部，通常需要处理一些自动化运行的任务，通常会采用系统自带的crontable的定时任务完成
- 但是，很多情况下，是多个作业，彼此先后执行，共同完成任务。在这样的情况下，定时任务存在两个明显的问题：
  - 浪费了大量的系统等待时间
  - 假设两个作业，第一个作业必须在第二个作业前运行，如果第二个作业先运行，就会有灾难性的后果，对于定时任务而言，解决任务这样两个作业优先级的问题是只能把任务一的运行时间安排在二之前，不能完全满足前面的假设，但是对于作业调度器而言，安排作业的优先级，是最基本的功能，简直是小Case
- 常见作业调度系统：Condor、LSF、PBS(PBS Pro、OpenPBS、TORQUE、曙光GridView、浪潮TSJM、联想LJRS)、Maui、Moab、SGE(Oracle Grid Engine)、SLURM

TORQUE和Maui

- 主页：http://www.adaptivecomputing.com/products/open-source/
- TORQUE资源管理器：
  - 2.5.x：不再发展，如不需要NUMA，建议选择
  - 3.0.x：在2.5系列基础上增加了对NUMA架构的支持，不建议使用
  - 4.0.x：增强千万亿次系统的支持
  - 4.1.x：针对Cray系统，不建议使用
  - 4.2.x：增加对Intel Xeon Phi的支持
- Maui集群调度：可与多种资源管理器配合，调度策略优，如TORQUE，负责作业调度

TORQUE服务端安装

- 解压缩：*tar␣xvf␣torque-2.5.12.tar.gz*
- 进入目录：*cd␣torque-2.5.12*
- 配置：*./configure␣--prefix=/opt/torque-2.5.12␣--with-scp*
- 编译：*make*
- 安装：*make␣install*
- 编辑*/etc/profile.d/torque.sh*：

```
TORQUE=/opt/torque-2.5.12
if [ "id -u" -eq 0 ]; then
    PATH=$PATH:$TORQUE/bin:$TORQUE/sbin
else
    PATH=$PATH:$TORQUE/bin
fi
```

TORQUE服务端配置

- 在解压缩后的源文件目录*torque-2.5.12*下：*./torque.setup␣root*
- 添加计算节点的机器名与对应的核数（np），编辑*/var/spool/torque/server_priv/nodes*：

```
node1 np=12
node2 np=8
```

- 创建队列：
  - 生成server：*pbs_server␣-t␣create*
  - 设置队列：*qmgr*，输入下面Qmgr:后的内容，将设置一个默认队列dque：

```
Qmgr: create  queue dque queue_type=execution
Qmgr: set  server  default_queue=dque
Qmgr: set  queue dque  started =true
Qmgr: set  queue dque enabled=true
Qmgr: set  server  scheduling=true
```

TORQUE测试

- 终止：*qterm␣−t␣quick*
- 启动：*pbs_server*
- 查看队列：*qstat␣−q*
- 查看配置：*qmgr␣−c␣'p␣s'*
- 查看节点信息：*pbsnodes␣−a*
- 提交作业测试：*echo␣"sleep␣30"␣|␣qsub*

计算节点安装TORQUE

- 服务端：在解压缩后的目录*torque-2.5.12*下运行*make␣packages*，生成：
  - *torque-package-clients-linux-x86_64.sh*
  - *torque-package-devel-linux-x86_64.sh*
  - *torque-package-doc-linux-x86_64.sh*
  - *torque-package-mom-linux-x86_64.sh*
  - *torque-package-server-linux-x86_64.sh*
- 计算节点：将*torque-package-clients-linux-x86_64.sh*和*torque-package-mom-linux-x86_64.sh*复制到计算节点，然后执行：
  - *torque-package-clients-linux-x86_64.sh␣--install*
  - *torque-package-mom-linux-x86_64.sh␣--install*

计算节点配置TORQUE

- 设置主节点名：*echo␣"admin"␣>/var/spool/torque/server_name*
- 编辑*/var/spool/torque/mom_priv/config*

```
$pbsserver      admin     # note: hostname running pbs_server
$logevent       255       # bitmap of which events to log
$usecp admin:/home /home
```

- 设置环境变量*/etc/profile.d/torque.sh*:

```
TORQUE=/opt/torque-2.5.12
if [ "id -u" -eq 0 ]; then
    PATH="/usr/local/sbin :/ usr/ local /bin :/ usr/ sbin :/ usr/ bin :/ sbin :/ bin :"
    PATH=$PATH:$TORQUE/bin:$TORQUE/sbin
else
    PATH="/usr/local/bin :/ usr/ bin :/ bin :/ usr/ games"
    PATH=$PATH:$TORQUE/bin
fi
```

服务端设置TORQUE服务

在解压缩后的目录*torque-2.5.12*下的子目录contrib/init.d下有系统启动时启动对应服务所需的脚本

- 修改*contrib/init.d/pbs_server*:

```
PBS_DAEMON=/opt/torque-2.5.12/sbin/pbs_server
```

- 复制到*/etc/init.d/*: *cp␣contrib/init.d/pbs_server␣/etc/init.d/*
- 设置系统启动时自启动:
  chkconfig␣--add␣pbs_server
  chkconfig␣pbs_server␣on
- 重启服务: *service␣pbs_server␣restart*

客户端设置TORQUE服务

- 修改*contrib/init.d/pbs_mom*:

```
PBS_DAEMON=/opt/torque-2.5.12/sbin/pbs_mom
```

- 复制到*/etc/init.d/*: *cp␣contrib/init.d/pbs_mom␣/etc/init.d/*
- 设置系统启动时自启动:
  chkconfig␣--add␣pbs_mom
  chkconfig␣pbs_mom␣on
- 重启服务: *service␣pbs_mom␣restart*

Maui安装

- Maui比TORQUE自带的集群调度器更好
- Maui只在服务端安装，计算节点无需安装
  - 解压缩：*tar␣xvf␣maui-3.3.1.tar.gz*
  - 进入目录：*cd␣maui-3.3.1*
  - 配置：

    ./configure␣--prefix=/opt/maui-3.3.1␣--with-pbs=/opt/torque-2.5.12
  - 编译：*make*
  - 安装：*make␣install*

服务节点上配置Maui

- 编辑*/usr/local/maui/maui.cfg*:

```
SERVERHOST            admin
# primary admin must be first in list
ADMIN1                root
# Resource Manager Definition
RMCFG[admin] TYPE=PBS@RMNMHOST@
RMTYPE[0] PBS
```

- 编辑*/etc/profile.d/maui.sh*设置环境变量:

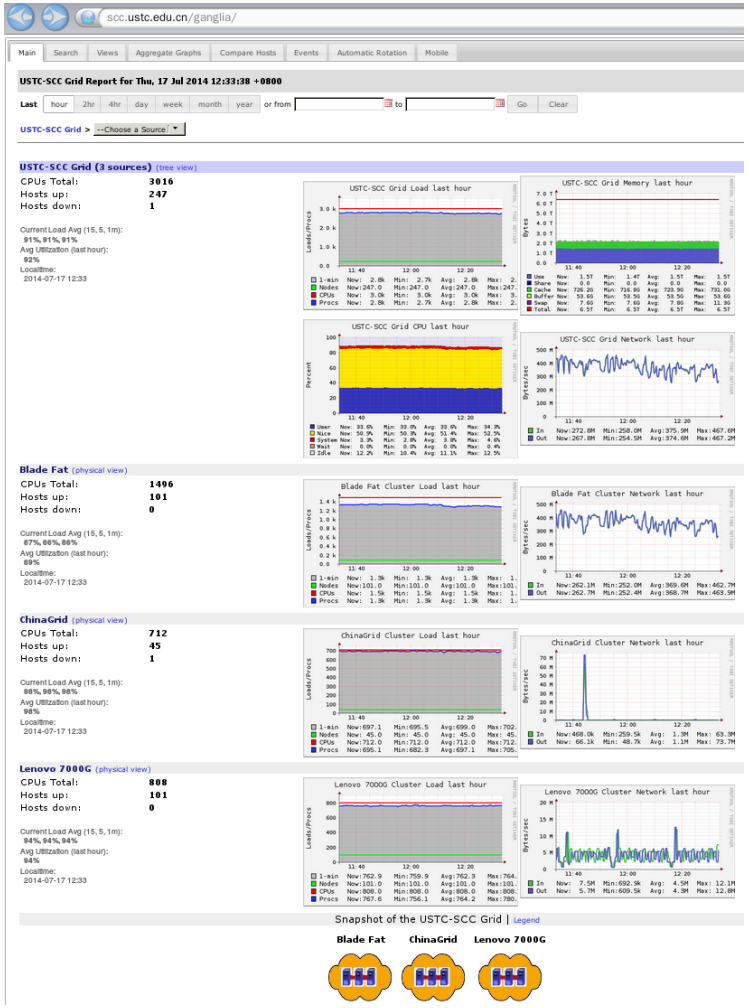
```
MAUI=/opt/maui-3.3.1
if [ "id -u" -eq 0 ]; then
  PATH=$PATH:$MAUI/bin:$MAUI/sbin
else
  PATH=$PATH:$MAUI/bin
fi
```

启动Maui

- 注意不要在服务节点上启动*pbs_sched*，如已启动，则先终止
- 启动Maui：*/opt/maui-3.3.1/sbin/maui*
- 设置系统启动时自启动，在*/etc/rc.local*中添加：

```
/opt/maui-3.3.1/sbin/maui
```

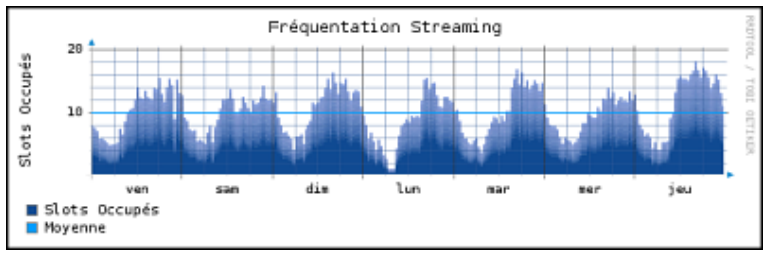
集群监控Ganglia



http://scc.ustc.edu.cn/ganglia/

集群监控Ganglia

- Ganglia：
  - 主要是用来监控系统性能的软件，如：cpu、mem、硬盘利用率、I/O负载、网络流量情况等
  - 通过曲线很容易见到每个节点的工作状态，对合理调整、分配系统资源，提高系统整体性能起到重要作用
  - Ganglia是分布式的监控系统，有两个Daemon：
    - 客户端Ganglia Monitoring Daemon(gmond)
    - 服务端Ganglia Meta Daemon(gmetad)
  - Ganglia Web：Ganglia基于PHP开发和运行的web统计浏览程序
- 依赖于：
  - PHP：基于服务端来创建动态网站的脚本语言，可生成网站主页
  - RRDtool：系统存放和显示time-series（网络带宽、温度、人数、服务器负载等），且可绘出有用的图表用来显示处理的数据和数据密度



- 官方站点：
  - http://ganglia.sourceforge.net/

Ganglia安装

CentOS官方源不含Ganglia，需添加第三方epel源，参见：
http://lug.ustc.edu.cn/wiki/mirrors/help/epel

- 服务端：*yum␣-y␣install␣ganglia-gmetad␣ganglia-gmond␣ganglia-web*
- 客户端：*yum␣-y␣install␣ganglia-gmond*

Ganglia客户端配置 I

- 编辑*/etc/ganglia/gmond.conf*:

```
cluster {
  name = "mycluster1" # 设置源，与服务端一致
  owner = "HMLi"
  latlong = "unspecified"
  url = "unspecified"
}
host {
  location = "5,0,2 # 节点位置，各节点不同，格式为 Rack, Rank and Plane
}
udp_send_channel {
  #bind_hostname = yes # Highly recommended, soon to be default.
                       # This option tells gmond to use a source address
                       # that resolves to the machine's hostname. Without
                       # this, the metrics may appear to come from any
                       # interface and the DNS names associated with
                       # those IPs will be used to create the RRDs.
  mcast_join = 239.2.11.71
  port = 8649
  ttl = 1
  mcast_if = eth0 # 发送信息的网卡
}
```

Ganglia客户端配置 II

```
udp_recv_channel {
  mcast_join = 239.2.11.71
  port = 8649
  bind = 239.2.11.71
  mcast_if = eth0 # 收集信息的网卡
}
```

Ganglia服务端配置 I

- 编辑*/etc/ganglia/gmond.conf*，内容与客户端基本一致：

```
cluster {
  name = "mycluster1" # 设置源
  owner = "HMLi"
  latlong = "unspecified"
  url = "unspecified"
}
host {
  location = "5,0,1" # 节点位置，各节点不同，格式为 Rack, Rank and Plane
}
udp_send_channel {
  #bind_hostname = yes # Highly recommended, soon to be default .
                       # This option tells gmond to use a source address
                       # that resolves to the machine's hostname. Without
                       # this , the metrics may appear to come from any
                       # interface and the DNS names associated with
                       # those IPs will be used to create the RRDs.
  mcast_join = 239.2.11.71
  port = 8649
  ttl = 1
 mcast_if = eth0 # 发送信息的网卡
}
udp_recv_channel {
```

Ganglia服务端配置 II

```
    mcast_join = 239.2.11.71
    port = 8649
    bind = 239.2.11.71
    mcast_if = eth0 # 收集信息的网卡
}
```

- 编辑*/etc/ganglia/gmetad.conf*:

```
data_source "mycluster1" localhost # 数据源名字，只收集客户端 cluster 段中 name 同名的
gridname "myGrid1" # 网格名
 all_trusted on # 允许收集其它节点的
```

Ganglia服务端配置 III

- 配置httpd，编辑*/etc/httpd/conf.d/ganglia.conf*:

```
Alias /ganglia /usr/share/ganglia

<Location /ganglia> # 可按照自己需要设置允许的 IP 访问范围
    Order allow,deny
    allow from all
   #Order deny,allow
   #Deny from all
   #Allow from 127.0.0.1
   #Allow from ::1
   # Allow from .example.com
</Location>
```

启动服务与测试

- 服务端：
  - 启动客户守护进程： *service␣gmond␣start*
  - 启动服务守护进程： *service␣gmetad␣start*
  - 启动httpd进程： *service␣httpd␣start*
  - 系统启动自启动客户守护进程： *chkconfig␣gmond␣on*
  - 系统启动自启动启动服务守护进程： *chkconfig␣gmetad␣on*
  - 系统启动自启动启动httpd守护进程： *chkconfig␣httpd␣on*
- 客户端：
  - 启动客户守护进程： *service␣gmond␣start*
  - 系统启动自启动启动服务守护进程： *chkconfig␣gmond␣on*
- 测试： *telnet␣admin␣8649␣|␣grep␣\<HOST*[9]应有类似输出：

```
<HOST NAME="node28" IP="10.0.0.28" REPORTED="1353728304" TN="1" TMAX="20" DMAX="0" LOCATION=
<HOST NAME="node46" IP="10.0.0.46" REPORTED="1353728305" TN="0" TMAX="20" DMAX="0" LOCATION=
<HOST NAME="node29" IP="10.0.0.29" REPORTED="1353728304" TN="1" TMAX="20" DMAX="0" LOCATION=
<HOST NAME="node47" IP="10.0.0.47" REPORTED="1353728305" TN="0" TMAX="20" DMAX="0" LOCATION=
<HOST NAME="node48" IP="10.0.0.48" REPORTED="1353728305" TN="0" TMAX="20" DMAX="0" LOCATION=
<HOST NAME="node49" IP="10.0.0.49" REPORTED="1353728305" TN="0" TMAX="20" DMAX="0" LOCATION=
Connection closed by foreign host.
```

- web访问： http://地址/ganglia/

[9] admin也可为其它节点名

联系信息

- 中国科大超级计算中心：
  - 电话：0551-63602248
  - 信箱：sccadmin@ustc.edu.cn
  - 主页：http://scc.ustc.edu.cn
  - 办公室：中国科大东区新图书馆一楼东侧126室
- 李会民：
  - 电话：0551-63600316
  - 信箱：hmli@ustc.edu.cn
  - 主页：http://hmli.ustc.edu.cn