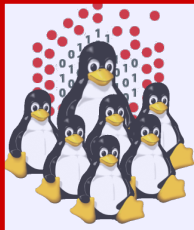


# Installation Procedures for Clusters



PART 2 – NETBOOT Environment  
and Troubleshooting

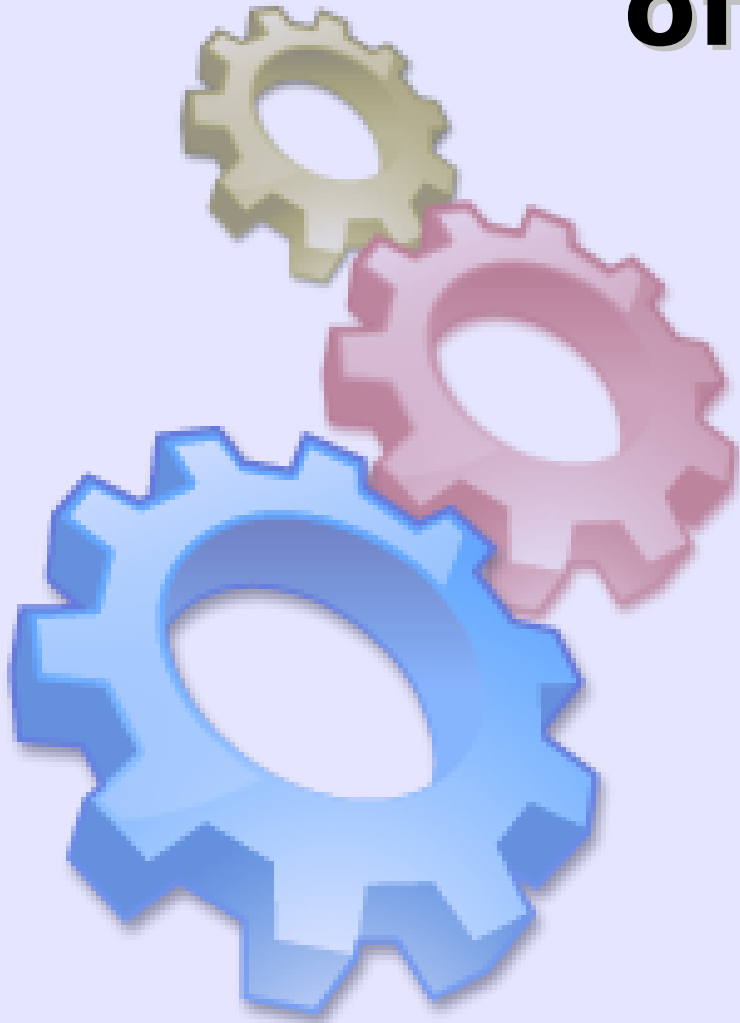




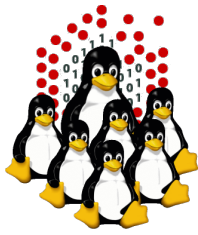
# Agenda

- Cluster Services
- Overview on Installation Procedures
- **Configuration and Setup of a NETBOOT Environment**
- **Troubleshooting**
- Cluster Management Tools
- Notes on Security
- Hands-on Laboratory Session

# Configuration and setup of NETBOOT services

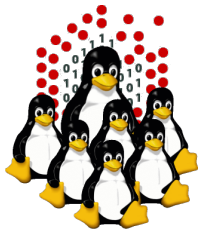


- **client setup**
  - PXE
  - BIOS
- **server setup**
  - DHCP
  - TFTP + PXE
  - NFS
  - Kickstart



# Setting up the client

- NIC that supports network booting (or etherboot)
- BIOS boot-sequence
  1. Floppy
  2. CD/DVD
  3. USB/External devices
  4. NETWORK
  5. Local Hard Disk
- Information gathering (client MAC address)
  - documentation (don't rely on this)
  - motherboard BIOS (if on-board)
  - NIC BIOS, initialization, PXE booting (need to monitor the boot process)
  - network sniffer (suitable for automation)



# Collecting MAC addresses

```
# tcpdump -c1 -i any -qtep port bootpc and port  
bootps and ip broadcast
```

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode

listening on any, link-type LINUX\_SLL (Linux cooked), capture size 96 bytes

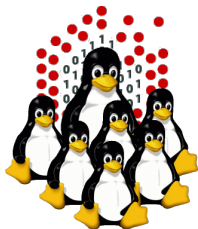
```
B 00:30:48:2c:61:8e 592: IP 0.0.0.0.bootpc >  
255.255.255.255.bootps: UDP, length 548
```

1 packets captured

1 packets received by filter

0 packets dropped by kernel

(see `/etc/services` for details on ports assignment)



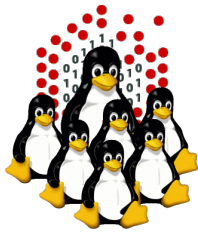
# Setting up DHCP

- It's a protocol that allows the dynamic configuration of the network settings for a client
- We need DHCP software for both the server and the clients (PXE implements a DHCP client internally)
- Steps needed
  - DHCP server package
  - DHCP configuration
  - client configuration
  - a TFTP server to supply the PXE bootloader
  - avoid conflicts - ignore queries from external networks (empty subnet declaration) or interfaces (dhcpd command line arguments)

```
ddns-update-style    none;
ddns-updates         off;
authoritative;
deny unknown-clients;

# cluster network
subnet 10.10.0.0 netmask 255.255.0.0 {
    option domain-name            "cluster.network";
    option domain-name-servers    10.10.0.1;
    option ntp-servers            10.10.0.1;
    option subnet-mask           255.255.0.0;
    option broadcast-address      10.10.255.255;
    # TFTP server
    next-server                   10.10.0.1;
    # NBP
    filename                      "/pxe/pxelinux.0";
    default-lease-time            -1;
    min-lease-time                864000;
}

# client section
host node01.cluster.network {
    hardware ethernet            00:30:48:2c:61:8e;
    fixed-address                10.10.1.1;
    option host-name              "node01";
}
```



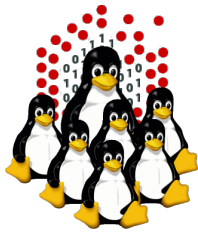
# Setting up DHCP

```
ddns-update-style none;  
ddns-updates off;  
authoritative;  
deny unknown-clients;
```

```
# cluster network  
subnet 10.10.0.0 netmask 255.255.0.0 {  
    option domain-name "cluster.network";  
    option domain-name-servers 10.10.0.1;  
    option ntp-servers 10.10.0.1;  
    option subnet-mask 255.255.0.0;  
    option broadcast-address 10.10.255.255;  
    # TFTP server  
    next-server 10.10.0.1;  
    # NBP  
    filename "/pxe/pxelinux.0";  
    default-lease-time -1;  
    min-lease-time 864000;  
}
```

```
# client section  
host node01.cluster.network {  
    hardware ethernet 00:30:48:2c:61:8e;  
    fixed-address 10.10.1.1;  
    option host-name "node01";  
}
```

Parameters starting with the `option` keyword correspond to actual DHCP options, while parameters that do not start with the `option` keyword either control the behavior of the DHCP server or specify client parameters that are not optional in the DHCP protocol.  
(man dhcpd.conf)



# TFTP and PXE

- What is TFTP

- Trivial File Transfer Protocol: is a simpler, faster, session-less and “unreliable” (based on UDP) implementation of the File Transfer Protocol;
- lightweight and simplicity make it the preferred way to transfer small files to/from network devices.

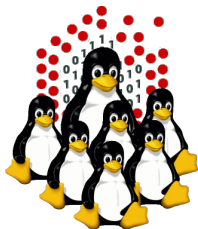
- What is PXE

- Pre-boot eXecution Environment, API burned-in into the PROM of the NIC
- provides a light implementation of some protocols (IP, UDP, DHCP, TFTP)

- What we need

- *tftp-server*, enabled as stand-alone daemon or through (x)inetd
- *pxelinux.0* from *syslinux* package (and *system-config-netboot*)
- the kernel (*vmlinuz*) and the initial ramdisk (*initrd.img*) from the installation CD
- a way to handle the node configuration file (<HEXIP>)
  - through TFTP
  - daemon on the server waiting for a connection from the installed node or *port-knocking*
  - CGI or PHP script (requires a web server)
  - directory exported via NFS





# PXE client configuration

configuration fall-back (MAC -> HEXIP -> default)  
paths relative to /tftpboot/pxe/pxelinux.cfg/

/tftpboot/pxe/pxelinux.cfg/default

```
prompt 1
timeout 100

display /pxelinux.cfg/bootmsg.txt

default local

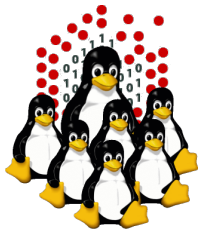
label local
    LOCALBOOT 0

label install
    kernel vmlinuz
    append vga=normal selinux=0 network ip=dhcp \
    ksdevice=eth0 ks=nfs:10.1.0.1:/distro/ks/nodes.ks \
    load_ramdisk=1 prompt_ramdisk=0 ramdisk_size=16384 \
    initrd=initrd.img

label memtest
    kernel memtest
```

```
/00000000-0000-0000-0000-0030481f8a97 # UUID
/01-00-30-48-2c-61-8e # MAC address
/0A0A0101 # 10.10.1.1 (IP ADDRESS)
/0A0A010 # 10.10.1.0-10.10.1.15
/0A0A01 # 10.10.1.0-10.10.1.255
/0A0A0 # 10.10.0.0-10.10.15.255
/0A0A # 10.10.0.0-10.10.255.255
/0A0 # 10.0.0.0-10.15.255.255
/0A # 10.0.0.0-10.255.255.255
/0 # 0.0.0.0-15.255.255.255
/default # nothing matched
```

Note: here '\' means that the line  
continue, but it **MUST** be actually  
written as a single line.

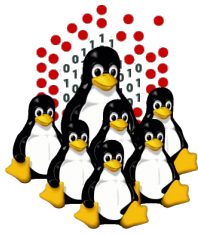


# Setting up the TFTP tree

- Populating the filesystem tree...

```
/
|-- tftpboot/
    |-- pxe/
        |-- vmlinuz
        |-- initrd.img
        |-- memtest
        |-- pxelinux.0
        |-- pxelinux.cfg/
            |-- 0A0A0101
            |-- bootmsg.txt
            |-- default -> default.local
            |-- default.install
            |-- default.local
```

- **Permissions:** world readable for “get”; writable flags and ownerships depend on how the <HEXIP> file is handled (tftp, web, nfs, daemon, ...)
  - tftp: needs world writable <HEXIP> file (for “put”)
  - nfs: directory exported (and mounted) as RW
  - daemon: ownerships and permissions depend on the UID
  - web: ownerships for the web server user

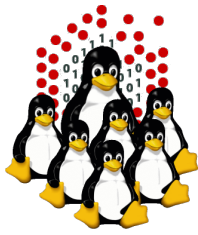


# Setting up NFS

- Create a local repository for RPM packages
- Copy the RPMs from the installation CDs/DVD or the ISO image(s), or just export the loop-mounted iso image(s)
- Export the repository to the cluster internal network
- Export the directory on which the kickstart resides
- Start/restart NFS service (or just “`exportfs -r`”)

Configuration sample (`/etc/exports`)

```
/distro          10.10.0.0/16(ro,root_squash)
```

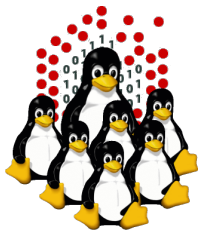


# Setting up KICKSTART

- Part of RedHat installation suite (Anaconda)
- Based on RPM packages and supported by all RH-based distros
- Allows non-interactive batch installation
- `system-config-kickstart` permit to create a template file

The kickstart configuration file, among other things, allows:

- ➔ network setup
- ➔ HD partitioning
- ➔ basic system configuration
- ➔ packages selection (`%packages`)
  - @<package-group>
  - <package> (add)
  - <package> (remove)
- ➔ pre-installation operations (`%pre`)
  - HW setup
  - specific configuration
- ➔ post-installation operations (`%post`)
  - post configuration, customization
  - stop the automated installation procedure



# KICKSTART example

/distro/ks/nodes.ks

```
install
nfs --server=10.10.0.1 --dir=/distro/WB4/
text
lang en_US
langsupport --default=en_US en_US
keyboard us
network --device eth0 --bootproto dhcp
network --device eth1 --bootproto dhcp
...
bootloader --location=mbr --append selinux=0
clearpart --all --initlabel
zerombr yes
part swap --size=4096 --asprimary
part / --fstype "ext3" --size=4096 --asprimary
part /local_scratch --fstype "ext3" --size=100 --grow
...
skipx

%packages --resolvedeps
ntp
openssh
openssh-server
-sendmail
...



```

%pre
hdparm -d1 -u1 /dev/hda 2>&1

```


```

```
%post --nochroot
cp /tmp/ks.cfg /mnt/sysimage/root/install-ks.cfg
cp /proc/cmdline /mnt/sysimage/root/install-cmdline

%post --interpreter=/bin/bash

exec 1>/root/post.log
exec 2>&1
set -x
export MASTER=10.10.0.1

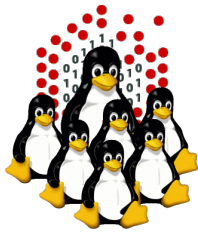
tftp_get() { tftp $MASTER -v -c get $1 $2 ; }
tftp_put() { tftp $MASTER -v -c put $1 $2 ; }

ip_to_hex() {
    /sbin/ip addr show dev $1 |
    sed -r '\|\s+inet\s([\^/]+)/.*\|!d;s/\^1/' |
    awk -F. '{printf("%02X%02X%02X%02X", $1, $2, $3, $4);}'
}

for eth in eth0 eth1 eth2
do
    HEX=`ip_to_hex $eth`
    test "x$HEX" != "x" && break
done

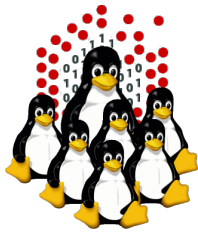
tftp_get /pxe/pxelinux.cfg/default.local /tmp/$HEX
tftp_put /tmp/$HEX /pxe/pxelinux.cfg/$HEX
```





# System logs

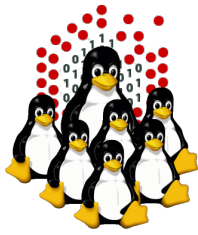
- Check system logs for:
  - DHCP negotiation (DISCOVER, OFFER, REQUEST, ACK/NACK)
  - DHCP leases (`/var/lib/dhcp/dhcpd.leases`)
  - TFTP transfers (enable verbose logging with `-vvv`)
  - denied/successful NFS mount (`showmount`)
  - connections rejected by server(s) configuration, *TCPwrapper*, firewall rules



# Network traffic analysis

- Sniff the network activity with:
  - tcpdump
  - wireshark/ethereal (tshark/tethereal)
- Look for:
  - client's ethernet MAC address (any packet sent by the node)
  - DHCP negotiation (DISCOVER, REQUEST, NACK)
  - TFTP UDP traffic
  - (NFS traffic)





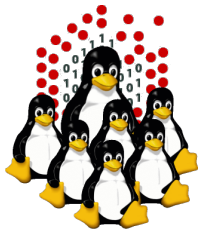
# Client virtual consoles (anaconda)

## FIRST STAGE

- CTRL+ALT+F1      BOOT, TEXTUAL CONFIGURATION
- CTRL+ALT+F2,F3      LOGS

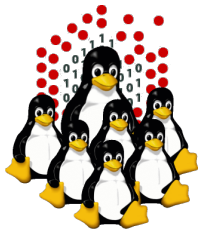
## SECOND STAGE

- CTRL+ALT+F1      LAUNCH X, REBOOT LOGS
- CTRL+ALT+F2      **SHELL**
- CTRL+ALT+F3,F4,F6      LOGS, DEBUG
- CTRL+ALT+F7      GRAPHICAL CONFIGURATION (X)



## At runtime ...

- use NTP to synchronize time/date cluster-wide
- collect logs on the masternode
  - ➔ masternode: `syslogd -r -x -m0`
  - ➔ nodes: `/etc/syslog.conf:*.* @masternode`
- logcheck/logwatch
- logrotate
- `tail -f ...`
- `grep ...`

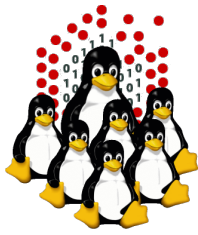


# That's All Folks!



```
( questions ; comments ) | mail -s uheilaaa baro@democritos.it
```

```
( complaints ; insults ) &>/dev/null
```



# REFERENCES AND USEFUL LINKS

## Cluster Toolkits:

- OSCAR – Open Source Cluster Application Resources  
<http://oscar.openclustergroup.org/>
- NPACI Rocks  
<http://www.rocksclusters.org/>
- Scyld Beowulf  
<http://www.beowulf.org/>
- CSM – IBM Cluster Systems Management  
<http://www.ibm.com/servers/eserver/clusters/software/>
- xCAT – eXtreme Cluster Administration Toolkit  
<http://www.xcat.org/>
- Warewulf/PERCEUS  
<http://www.warewulf-cluster.org/> <http://www.perceus.org/>

## Installation Software:

- SystemImager <http://www.systemimager.org/>
- FAI <http://www.informatik.uni-koeln.de/fai/>
- Anaconda/Kickstart <http://fedoraproject.org/wiki/Anaconda/Kickstart>

## Management Tools:

- openssh/openssl  
<http://www.openssh.com>  
<http://www.openssl.org>
- C3 tools – The Cluster Command and Control tool suite  
<http://www.csm.ornl.gov/torc/C3/>
- PDSH – Parallel Distributed SHell  
<https://computing.llnl.gov/linux/pdsh.html>
- DSH – Distributed SHell  
<http://www.netfort.gr.jp/~dancer/software/dsh.html.en>
- ClusterSSH  
<http://clusterssh.sourceforge.net/>
- C4 tools – Cluster Command & Control Console  
<http://gforge.escience-lab.org/projects/c-4/>

## Monitoring Tools:

- Ganglia <http://ganglia.sourceforge.net/>
- Nagios <http://www.nagios.org/>
- Zabbix <http://www.zabbix.org/>

## Network traffic analyzer:

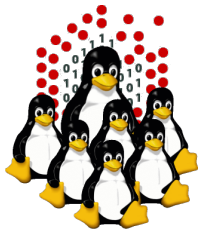
- tcpdump <http://www.tcpdump.org>
- Wireshark <http://www.wireshark.org>

## UnionFS:

- Hopeless, a system for building disk-less clusters  
<http://www.evolware.org/chri/hopeless.html>
- UnionFS – A Stackable Unification File System  
<http://www.unionfs.org>  
<http://www.fsl.cs.sunysb.edu/project-unionfs.html>

## RFC: (<http://www.rfc.net>)

- RFC 1350 – The TFTP Protocol (Revision 2)  
<http://www.rfc.net/rfc1350.html>
- RFC 2131 – Dynamic Host Configuration Protocol  
<http://www.rfc.net/rfc2131.html>
- RFC 2132 – DHCP Options and BOOTP Vendor Extensions  
<http://www.rfc.net/rfc2132.html>
- RFC 4578 – DHCP PXE Options  
<http://www.rfc.net/rfc4578.html>
- RFC 4390 – DHCP over Infiniband  
<http://www.rfc.net/rfc4390.html>
- PXE specification  
<http://www.pix.net/software/pxeboot/archive/pxespec.pdf>
- SYSINUX <http://syslinux.zytor.com/>



# Some acronyms...

**HPC** – High Performance Computing

**OS** – Operating System

**LINUX** – LINUX is not UNIX

**GNU** – GNU is not UNIX

**RPM** – RPM Package Manager

**CLI** – Command Line Interface

**BASH** – Bourne Again SHell

**PERL** – Practical Extraction and Report Language

**PXE** – Preboot Execution Environment

**INITRD** – INITIAL RamDisk

**NFS** – Network File System

**SSH** – Secure SHell

**LDAP** – Lightweight Directory Access Protocol

**NIS** – Network Information Service

**DNS** – Domain Name System

**PAM** – Pluggable Authentication Modules

**LAN** – Local Area Network

**WAN** – Wide Area Network

**IP** – Internet Protocol

**TCP** – Transmission Control Protocol

**UDP** – User Datagram Protocol

**DHCP** – Dynamic Host Configuration Protocol

**TFTP** – Trivial File Transfer Protocol

**FTP** – File Transfer Protocol

**HTTP** – Hyper Text Transfer Protocol

**NTP** – Network Time Protocol

**NIC** – Network Interface Card/Controller

**MAC** – Media Access Control

**OUI** – Organizationally Unique Identifier

**API** – Application Program Interface

**UNDI** – Universal Network Driver Interface

**PROM** – Programmable Read-Only Memory

**BIOS** – Basic Input/Output System

**SNMP** – Simple Network Management Protocol

**MIB** – Management Information Base

**OID** – Object Identifier

**IPMI** – Intelligent Platform Management Interface

**LOM** – Lights-Out Management

**RSA** – IBM Remote Supervisor Adapter

**BMC** – Baseboard Management Controller