Diskless Linux Clusters

Ciro Cattuto

ciro.cattuto@pg.infn.it

Outline:

- Why diskless operation?
- Issues with diskless nodes
- Our choices
- Configuring the Linux kernel
- The boot sequence
- Pros and cons

Why diskless operation?

Cost motivations

Why diskless operation?

Cost motivations

OSCAR is too simple! ;-)

Why diskless operation?

Cost motivations

OSCAR is too simple! ;-)

Single System Image (SSI) approach to clustering

- giving users and applications a unified cluster image
- requires integration at multiple levels: kernel, filesystem, namespace, . . .
- . . . but it's not there yet!

Why diskless operation?

Cost motivations

OSCAR is too simple! ;-)

Single System Image (SSI) approach to clustering

- giving users and applications a unified cluster image
- requires integration at multiple levels: kernel, filesystem, namespace, . . .
- . . . but it's not there yet!

A single root filesystem is the first step towards SSI:

Manageability

Flexibility

Why diskless operation?

Cost motivations

OSCAR is too simple! ;-)

Single System Image (SSI) approach to clustering

- giving users and applications a unified cluster image
- requires integration at multiple levels: kernel, filesystem, namespace, . . .
- . . . but it's not there yet!

A single root filesystem is the first step towards SSI:

Manageability

Flexibility

Sometimes, even when you do have disks,
you don't want to touch them!

Issues with diskless nodes

- How do I boot?
  - *Floppy*
  - *Network (PXE, etherboot)*
- Where is my root filesystem?
  - *Network file system (NFS)*
- Where do I swap to?
  - *Just don't.*
  - *Swap over the network*

Issues with diskless nodes

- How do I boot?
  - *Floppy*
  - *Network (PXE, etherboot)*

- Where is my root filesystem?
  - *Network file system (NFS)*

- Where do I swap to?
  - *Just don't.*
  - *Swap over the network*

Moreover, if we want a single root filesystem for all nodes:

- Where is my identity?
  - *Created on the fly*
  - *Stored remotely*

Requirements and Choices

Our requirements for system organization:

- Dedicated system (no PBS, . . .)
- Easily reconfigurable at the system level:
  - ▷ *System libraries*
  - ▷ *Kernel*
  - ▷ *Running set of daemons*
  - ▷ *Cluster-wide user environment*

Our choices:

- Root filesystem over NFS, read-only mounted
  - ▷ *root filesystem based on the Debian distribution*
  - ▷ *the root filesystem exported to nodes is maintained using standard Debian tools (dpkg, apt-get)*
- GRUB bootloader
  - ▷ *integrates a BOOTP/DHCP client*
  - ▷ *can fetch and execute scripts specifying a boot sequence*
  - ▷ *we boot from floppies or PXE*
- Small local ramdisk, for volatile write access
  - ▷ `/tmp` *has to be writable! (think of lock files . . .)*
- Devfs virtual filesystem
  - ▷ *No device inode lookups over the network*
  - ▷ *No* `/dev` *files on the exported root filesystem*
- NFS-shared `/beowulf` filesystem
  - ▷ *holds the application libraries (MPICH, LAM, FFTW, . . .)*
  - ▷ *holds home directories*

Compute nodes

A closer look at one node . . .

```
cattuto@node10:~$ df -aTm
Filesystem        Type   1M-blocks   Used Available Use% Mounted on
/dev/root          nfs        5613   1341      3987  26% /
none               devfs         0      0         0    -  /dev
proc               proc          0      0         0    -  /proc
/dev/ram0          ext2          8      1         7   1% /ramdisk
node00:/beowulf    nfs       10199   8524      1676  84% /beowulf
```

A closer look at one node . . .

```
cattuto@node10:~$ df -aTm
Filesystem        Type    1M-blocks    Used Available Use% Mounted on
/dev/root          nfs         5613    1341      3987  26% /
none               devfs          0       0         0    -  /dev
proc               proc           0       0         0    -  /proc
/dev/ram0          ext2           8       1         7   1% /ramdisk
node00:/beowulf    nfs        10199    8524      1676  84% /beowulf
```

```
cattuto@node10:~$ ls -l /tmp
lrwxrwxrwx  1 root  root    12 Jan 30 13:48 /tmp -> /ramdisk/tmp/
```

A closer look at one node . . .

```
cattuto@node10:~$ df -aTm
Filesystem        Type   1M-blocks   Used Available Use% Mounted on
/dev/root          nfs        5613   1341      3987  26% /
none               devfs         0      0         0    -  /dev
proc               proc          0      0         0    -  /proc
/dev/ram0          ext2          8      1         7   1% /ramdisk
node00:/beowulf    nfs       10199   8524      1676  84% /beowulf
```

```
cattuto@node10:~$ ls -l /tmp
lrwxrwxrwx  1 root  root    12 Jan 30 13:48 /tmp -> /ramdisk/tmp/
```

```
cattuto@node10:~$ ls -l /ramdisk
drwxr-xr-x  3 root  root  1024 Feb  8 15:02 etc/
drwxrwxrwt  5 root  root  1024 Feb 14 00:17 tmp/
drwxr-xr-x  9 root  root  1024 Feb  6 18:08 var/
cattuto@node10:~$
```

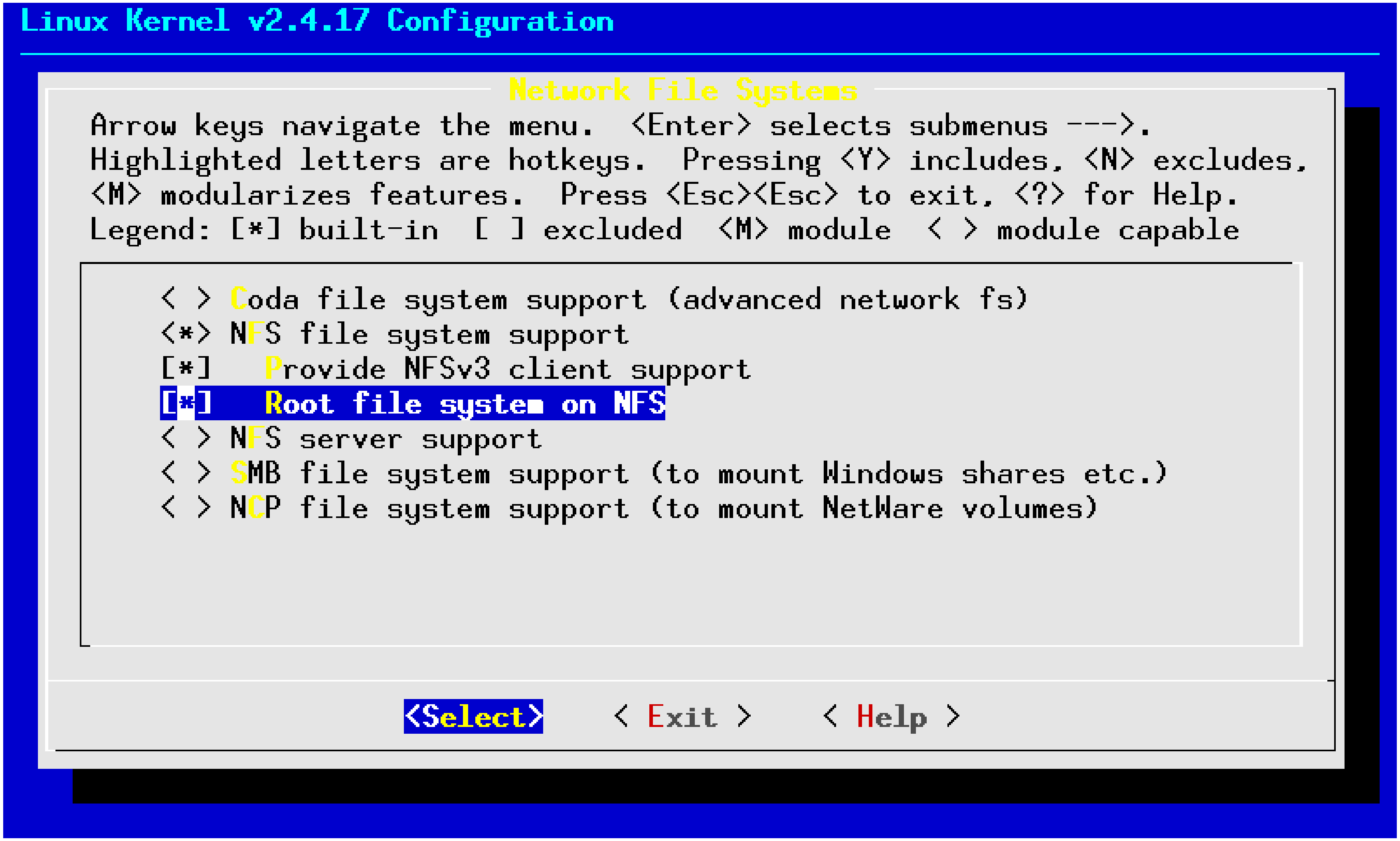
Kernel configuration (1)

```
Linux Kernel v2.4.17 Configuration
                         Networking options
  Arrow keys navigate the menu.  <Enter> selects submenus --->.
  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,
  <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help.
  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable
        ^(-)
        <*> Unix domain sockets
        [*] TCP/IP networking
        [ ]   IP: multicasting
        [ ]   IP: advanced router
        [*]   IP: kernel level autoconfiguration
        [*]     IP: DHCP support (NEW)
        [*]     IP: BOOTP support (NEW)
        [ ]     IP: RARP support (NEW)
        < >   IP: tunneling
        < >   IP: GRE tunnels over IP
        v(+)

              <Select>    < Exit >    < Help >
```

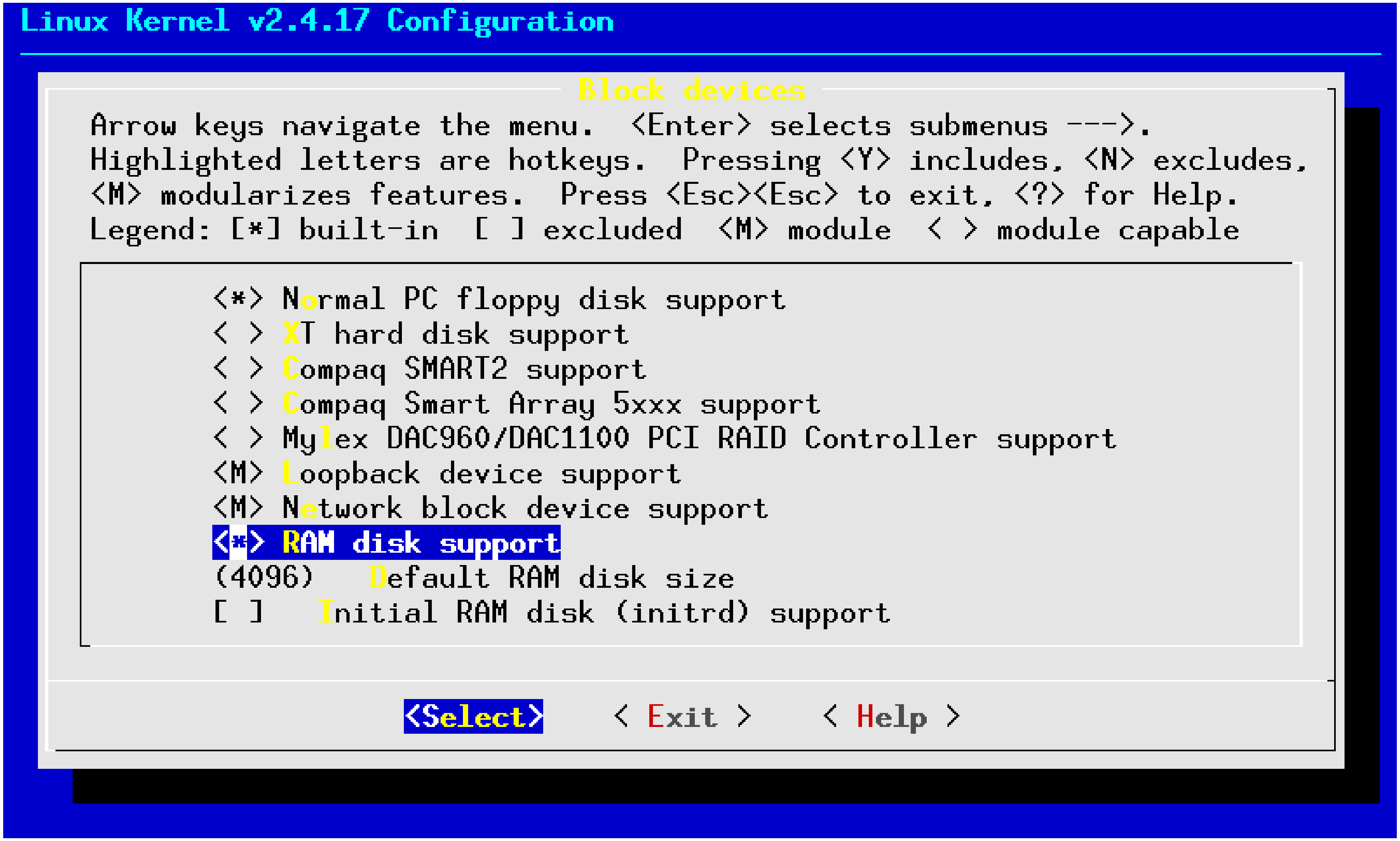
→ *enable boot-time automatic IP configuration*

Kernel configuration (2)



→ *enable NFS support and NFSroot functionality*

Kernel configuration (3)



→ *enable ramdisk support*

The boot sequence

1. Power on. GRUB is loaded from the PXE ROM or boot floppy

2. GRUB probes the NIC and sends out
   a DHCP query to configure the network

3. GRUB downloads (via TFTP) a boot script from the DHCP server

4. The boot script fetches a kernel image
   and fires it up, passing it proper parameters

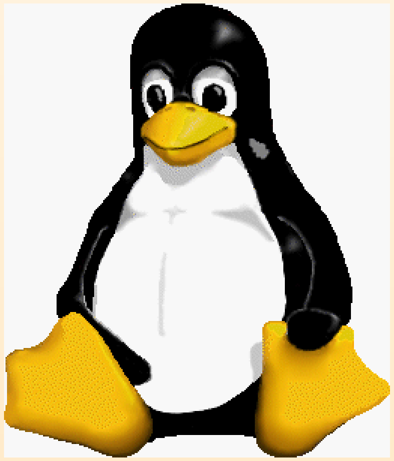
The boot sequence

1. Power on. GRUB is loaded from the PXE ROM or boot floppy
2. GRUB probes the NIC and sends out
   a DHCP query to configure the network
3. GRUB downloads (via TFTP) a boot script from the DHCP server
4. The boot script fetches a kernel image
   and fires it up, passing it proper parameters
5. The Linux kernel boots and the IP autoconfiguration code
   sends out a DHCP query to configure the NIC
6. The root filesystem is mounted over NFS (read-only)
   from the DHCP server
7. System initialization begins.

8. Devfs is mounted

9. The ramdisk is created, populated and mounted

8. Devfs is mounted
9. The ramdisk is created, populated and mounted
10. Boot scripts complete local configuration (on ramdisk)
11. Remote `/beowulf` filesystem gets mounted
12. Per-node configuration is performed (if any)
13. System is ready!



A glance at the server side . . .

```
virgo-bwulf:/beowulf/boot# ls -l
  1 root  root 827361 Nov 23 16:18 bzImage-2.4.17
  1 root  root 921385 Nov 18 16:13 bzImage-2.4.17-mosix
  1 root  root    151 Jan 29 21:34 install.lst
  1 root  root    232 Jan 29 21:34 local.lst
  1 root  root    139 Jan 29 21:34 node-rw.lst
  1 root  root    102 Jan 29 21:34 node.lst
  1 root  root     11 Feb 14 00:49 node10.lst -> install.lst
```

- At boot time, GRUB configures the network via DHCP
- Then, each node attempts to download and execute a GRUB boot script:
  - ▷ *if there is a node-specific script, go for it*
  - ▷ *otherwise, use the generic* `node.lst`
- The boot script downloads the right kernel and starts it
  - ▷ *the kernel command line can be manipulated by GRUB*
- Kernel and boot sequence of all the nodes can be controlled by changing files and symlinks in `/beowulf/boot` - and only that.

Pros and Cons

Pros:

- Manageability
  - ▷ *Single point of control for kernel image and boot sequence of all nodes*
  - ▷ *Single point of control for system libraries*
  - ▷ *Single point of control for application libraries (`/beowulf`)*
  - ▷ *Single point of control for user environment (`/beowulf/env`)*
  - ▷ *Chrooted operation on node root filesystem, using Debian tools*
  - ▷ *Compute nodes share an identical namespace*
- Flexibility
  - ▷ *Nodes need not have the same running set of daemons*
  - ▷ *Local disks, if present, can be automatically partitioned and populated (useful for local swapping, PVFS, . . .)*
- No loss of performance wrt nodes with local installation (OSCAR style)

Cons:

- Scalability
  - ▷ *During boot, all nodes access the same NFS filesystem (then VFS caching takes over)*
  - ▷ *In practice: no noticeable slowdown of boot sequence was observed*

Other directions:

- Scyld: second generation Beowulf clusters
  - *User friendly cluster installation procedure*
  - *Diskless operation*
  - *Kernel modifications to support distributed process space (BProc)*
  - *Kernel lightweight facilities to start remote processes*
  - *Cluster control tools*

Other directions:

- Scyld: second generation Beowulf clusters
  - *User friendly cluster installation procedure*
  - *Diskless operation*
  - *Kernel modifications to support distributed process space (BProc)*
  - *Kernel lightweight facilities to start remote processes*
  - *Cluster control tools*

- Single System Image efforts

  "I really want to see the Compaq clustering code, the IBM DLM and OpenGFS in the 2.5 tree creating a real clustered Linux with true failover facilities. That will really open the door to the enterprise market." -Alan Cox (Nov 8th, 2001)

References

- GRUB
  http://www.gnu.org/software/grub/
- Linux NFS-Root mini-HOWTO
  http://www.linuxdoc.org/HOWTO/mini/NFS-Root.html
- Linux Diskless HOWTO
  http://www.linuxdoc.org/HOWTO/Diskless-HOWTO.html
- Devfs FAQs
  http://www.atnf.csiro.au/~rgooch/linux/docs/devfs.html
- Debian
  http://www.debian.org/
- Single System Image Clusters for Linux
  http://ssic-linux.sourceforge.net/
- Scyld
  http://www.scyld.com/

Acknowledgements

- INFM
- INFN/CNRS
- VIRGO Perugia Group
- Leone Bosi

This presentation has been prepared using Free Software tools only,
see http://www.cs.berkeley.edu/∼mdw/proj/texslides/ :-)