

NFS-Clients in SW- und Geräteentwicklung

- **Einleitung**
- **Entscheidungsprozeß**
 - **Historie DIMAX**
 - **Anforderungen an digitales D-WS**
 - **Betriebssystemauswahl**
- **Projektphase I**
 - **Server und Arbeitsplätze**
 - **diskless PPC Testsysteme**
- **Projektphase II**
 - **Arbeitsplätze als NFS-Clients**
 - **Prototypen als NFS-Clients**

Einleitung: Vorstellung

Axel Dittrich, SW-Engineer

Systemadministration für SW-Abteilung ca. 50% Zeit

AGFA Gevaert AG,

Laborgeräte München

Produkte

- **Großgeräte für Fotoindustrie**
- **Minilabs für Stundenshops**

Historie DIMAX

- **Entwicklungszeit ca. 1994-1999**
- **Leistung: 20000 Bilder/h**
- **HP-RT (Lynx OS Derivat) auf PA-RISC VME-Bus Karte**
- **X-Terminal für Grafikausgabe mit Motiv-GUI**
- **Hybride Belichtung: Film, Optik, Fotopapier + Kontrastausgleich mit SW-LCD**
- **DMD/DLP-Indexprintsystem mit WinNT**
- **32-Bit MIPS Karten für Signalverarbeitung + spezielle HW**
- **mehrere C167 CAN Controller für Mechanikbaugruppen**
- **Crossentwicklung unter HP-UX**
- **HP-UX Workstation, X-Terminal, Linux-PC's, WinNT mit Exceed**

Anforderungen



- 20000 Scans/h = 180 ms/Bild
- 4 Farben mit je 3k*2k*12 Bit CCD
- ca. 300 MB/s Raw Daten
- Bildverarbeitung in FPGA und DSP's zu JPEG Files
- Bildspeicher ca. 4h = 100 GB
- Netzdatenrate ca. 4-6 MB/s

- 20000 Prints/h = 180 ms/Bild
- Laserbelichter mit 300dpi
- Bildverarbeitung in DSP's
- Netzdatenrate ca. 4-6 MB/s
- Rückseitendruck
- automatischer Papierwechsler mit Schneiden und Kleben



Betriebssystemauswahl (1)

- **Serveranforderungen sowie "weiche" Echtzeit**
- **große Entwicklergruppe über mehrere Jahre**
- **Nutzung vorhandenen Know How's, weniger durch konkrete Wiederverwendung als durch Prinzipien und allgemeine Erfahrungen wegen großem Technologiesprung**
- **Auftragsstudien, Workshops, Produktdemos, Embedded Systems, 1. RT-Linux-Workshop in Wien**
- **eigene Untersuchungen und Tests mit QNX, LynxOS, Linux, RTAI-Linux, Low-Latency-Patches**
- **möglichst Support in der Nähe, zumindest Deutschland**
- **bevorzugt x86-PC-Technik**
- **cPCI wegen Bilddatenbus, längerer Liefergarantie, Qualität**

Betriebssystemauswahl (2)

- vollständige Java Unterstützung
- Bewertung von nutzbaren Tools je Plattform
- Journalized Filesystem wegen 100 GB Bilddaten
- Unterstützung SCSI und RAID (auch SW)

=> kein OS unterstützt(e) Server- und Echtzeitanforderungen

=> Aufteilung in mehrere Subsysteme

=> "Echtzeit"-System UNIX-artig mit POSIX-API wegen positiven Erfahrungen, Entscheidung für Linux

=> Serversystem Linux oder WinNT/2000, Entscheidung für Linux wegen Gleichartigkeit zu "Echtzeit"-System, tieferem UNIX Know How, Kosten ABER: politisch nicht einfach!

Phase I: Server und Arbeitsplätze

- **HP-UX Workstations von DIMAX Projekt, NIS**
- **Linux-Server als zentraler Fileserver, NFS+Samba**
- **Linux Arbeitsplatzrechner manuell installiert (meist geklont)**
- **neue SW-Pakete manuell installiert oder in zentrale NFS-Verzeichnisse, Automounter**
- **Konfigurationspflege durch NIS und SUP (SoftwareUpdateProtokoll) vereinfacht**

=> zunehmende Anzahl Linux-Arbeitsplatzrechner bedarf einfacherer Pflege, da zum Projektstart häufiger Wechsel/Test von Entwicklungs-Tools, Libraries oder auch Kernel- und Distributions-Versionen

Phase I: diskless PPC Systeme

- einzelne Testsysteme direkt am Hausnetz
- "Echtzeit"-System für Mechaniksteuerung über CAN
- IBM PPC405 200 MHz cPCI Karte mit 100MBit, 64 MB, CAN
- Prescan-SV-System auf FPGA Motherboard
- Motorola MPC860 50 MHz Steckkarte mit 10 MBit, 16 MByte
- diskless Boot mit ppcboot/u-boot im Flash
- ca. 15 s (PPC405) - 25 s (MPC860) bis RSH-Login-Prompt
- mounten /home und /projects Hierarchien
- Cross- oder lokale Entwicklung mit gcc/g++

Phase II: Arbeitsplätze als NFS-Clients (1)

- **Ziel: einfache Pflege der Linux Arbeitsplatz-PC's und trotzdem Ausnutzung der vorhandenen Hardware (RAM, CPU, Grafik)**
- **Ziel: hohe Flexibilität und schnelle Änderbarkeit**
- **Ziel: möglichst nah an verwendeter Distribution bleiben**
- **Ziel: möglichst viel von Servern readonly nutzen**

- **LTSP , Linux-X-Terminals und Net-PC's als Anregung**
- **positive NFS-Erfahrungen für /home, /projects, /usr/local, ...**
- **positive Erfahrungen mit diskless PPC's und X-Terminals**
- **ca. 2 Monate Entwicklungszeit, Einarbeitung Initrd, Kernelbuild, bootp, tftp, init-Scripte**

Phase II: Arbeitsplätze als NFS-Clients (2)

- **Filebasierter Ansatz mit wenigen Generator-Scripten, dadurch Pflege ähnlich Einzel-Installationen**
- **95% wird readonly vom Server gemountet, /bin, /lib, /sbin, /usr, /opt**
- **/etc und /var werden größtenteils readonly verlinkt**
- **lokale Disk nur für Swap, /tmp und Bootloader**
- **RAM-Disk für RW-Daten**
- **Template Verzeichnis für alle Clients**
- **Clientweise Neudefinition weniger Files (oft < 20)**
- **Clientdaten überschreiben Template-Defaults überschreiben Server-Defaults**

Phase II: Prototypen als NFS-Clients (3)

CLIENT Verzeichnis:

```
./dev
./dev/cdrom          (Softlink auf CDROM Device)
./dev/mouse         (Softlink auf Mouse Device)
./etc
./etc/HOSTNAME      (Client Name)
./etc/X11
./etc/X11/XF86Config.90.Screen (modularisiertes XF86Config wird per
./etc/X11/XF86Config.95.ServerLayout Script gebaut, Rest von Template)
./etc/fstab         (welche Partitionen hat der Client)
./etc/init.d
./etc/init.d/boot.local (spezielle "Hacks", hier z.B. für VMware)
./etc/init.d/boot.nfsclient (Client Anpassungen vor Start von init)
./etc/init.d/nmb    (Softlink zum Aktivieren, Default: dummy)
./etc/init.d/smb    (Softlink zum Aktivieren, Default: dummy)
./etc/passwd.local (spez. User, z.B. lroot)
./etc/samba
./etc/samba/smb-emu461.conf (Samba include File zum Default)
./etc/shadow.local (spez. User, z.B. lroot)
./etc/vmware
./etc/vmware/config (VMware braucht schreibbares File)
```

Phase II: Arbeitsplätze als NFS-Clients (4)

```
EXT2-fs warning: mounting unchecked fs, running e2fsck is recommended
UFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 84k freed
+++ BEGIN NFSCLIENT LINUXRC +++
+++ VERSION : linuxrc,v 1.12 2003/06/11 22:24:46 root Exp root $ +++
+++ LINUXRC CMDLINE: 3
proc on /proc type proc (rw)
+++ KERNEL CMDLINE: rw root=100 ramdisk_size=8192 NET=pcnet32 CLIENT=eip02 ASK=1
3 ROOT=pcls02:/
+++ Loading module pcnet32 ...
Using /lib/modules/2.4.21-nfsclient-SuSE82/kernel/drivers/net/mii.o
Symbol version prefix ''
Using /lib/modules/2.4.21-nfsclient-SuSE82/kernel/drivers/net/pcnet32.o
Symbol version prefix ''
pcnet32.c:v1.27a 10.02.2002 tsbogend@alpha.franken.de
PCI: Found IRQ 9 for device 00:10.0
pcnet32: PCnet/PCI II 79C970A at 0x1060, 00 50 56 20 05 90 assigned IRQ 9.
eth0: registered as PCnet/PCI II 79C970A
pcnet32: 1 cards_found.
+++ Starting dhcpd on dev eth0 ...
retry=10 TIMEOUT=5
dhcpd: MAC address = 00:50:56:20:05:90
dhcpd[13]: broadcasting DHCP_DISCOVER

dhcpd[13]: dhcpIPaddrLeaseTime=518400 in DHCP server response.
```

Phase II: Arbeitsplätze als NFS-Clients (5)

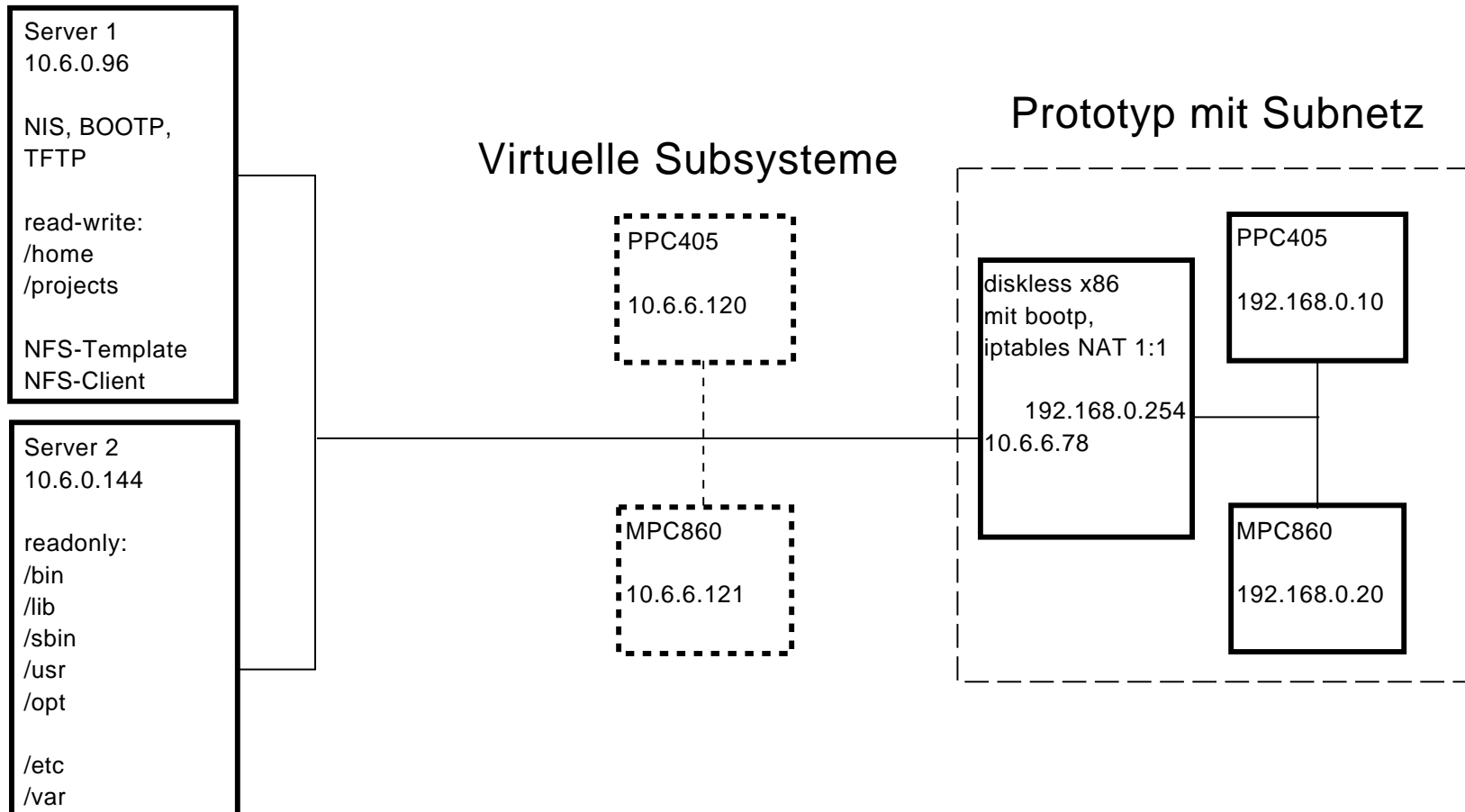
```
+++ NISSRV=pcls01
+++ NISPORT=3001
+++ ASK=1
+++ RUNLEVEL=3
+++ Mounting NFS tree ...
[slh in initrd, [blash from NFS-SERVER, [ilnit from NFS-SERVER b
+++ BEGIN /NFS/ALL/etc/init.d/boot.nfsclient +++
+++ Make dirs and links in / ...
+++ Mount /opt /usr /var from pcls02.fot.agfa.de ...
+++ Fillup /dev .. from ALL .. from CLIENT .. from ROOT .. done
+++ Fillup /etc .. from ALL .. from CLIENT .. from ROOT .. done
+++ Fillup /var .. from ALL .. from CLIENT .. done
+++ Fillup /root .. from ROOT .. done
+++ Mount modules dir pcls01:/srv/projects/nfsclient/lib/modules ...
+++ BEGIN /NFS/CLIENT/etc/init.d/boot.nfsclient +++
+++ Loading module ide-mod, ide-probe-mod ...
+++ Make client specific dirs ...
+++ STOP /NFS/CLIENT/etc/init.d/boot.nfsclient +++
+++ END /NFS/ALL/etc/init.d/boot.nfsclient +++
+++ Cleanup ramfs ...
+++ Copying dhcp leases and nis info ...
+++ Removing unnessecary files ...
+++ END NFSCLIENT LINUXRC +++

root@eip02:~#
```

Phase II: Prototypen als NFS-Clients (1)

- Ziel: einzelne SW-Prozesse und Baugruppen sollen am Prototyp verbunden und gemeinsam gestartet werden
 - Ziel: einfache Pflege, hohe Flexibilität und schnelle Änderbarkeit gerade zu Beginn der Prototypphase
 - Ziel: gemeinsame/wechselseitige Nutzung der Prototypen durch verschiedene Entwickler mit "ihren" SW-Versionen
 - Ziel: Dualboot Fähigkeit, vom Netz für Tests durch einzelne Entwickler, von Harddisk für Tests von offiziellen Referenzversionen
 - Ziel: volle Maschinenfunktionalität auch bei Netzboot
- => Hauptrechner mit Etherboot mit Bootmenü, bootp, iptables mit NAT 1:1,

Phase II: Prototypen als NFS-Clients (2)



Zusammenfassung

- **positive Erfahrungen nach Entscheidung für Linux als Basisplattform**
- **Server können mit zentral gepflegten NFS-Arbeitsplätzen gut ergänzt werden**
- **homogene Systemlandschaft auch für den Nutzer**
- **Leistung und Flexibilität direkt am Entwickler-Arbeitsplatz**
- **effiziente Nutzung auch für Prototypen, selbst mit eigenem internem Netzwerk und diskless Subsystemen**
- **bewährt insbesondere bei Systemwechsel von SuSE-7.2 zu SuSE-8.2 auf allen Servern, Arbeitsplätzen und Prototypen, selbst Parallelbetrieb war durch Bootmenü möglich**

Referenzen

- **Suchodoletz: "Die Netzstarter"**, Linux-Magazin 1/2003, S. 74ff
- **Linux Terminal Server Project:** <http://www.ltsp.org/>
- **Linux Diskless Client:** <http://ldc.goe.net/>
- **Softwareverteilung:** Linux-Magazin 7/2003, S.28ff
- **Pflege Linux-PC's:** Linux-Magazin 5/2003, S.33
- **Knoppix:** <http://www.knoppix.org/>
- **Etherboot:** <http://www.etherboot.org/>
- **Rdist:** <http://www.MagniComp.com/rdist/rdist.shtml>
- **Software Upgrade Protokoll:**
<http://packages.debian.org/stable/devel/sup>