

# Automatische Installationen mit FAI



Thomas Lange  
Institut für Informatik  
Universität zu Köln  
`lange@informatik.uni-koeln.de`

Heidelberg, Mai 2002

# Einleitung

- Die Inbetriebnahme (Planung, Kauf, Test, Installation, Wartung) eines Clusters braucht viel Zeit und Fachwissen
- Gute Installation von Betriebssystem und Anwendungen ist wichtig
- Schlechte Installation führt zu unbrauchbaren Rechnern
- Die Arbeiten in der Systemadministration muss skalieren !
- PC Hardware ist billig, schnell, aber oft defekt
- Beowulf Beispiel: 26×1GHz Athlon, 256MB RAM, 20GB disk, Fast Ethernet Switch, Kabel, Regal und DAT Laufwerk für 20.000 Euro Ende 2001 gekauft, Institut für physikalische Chemie

# Was ist FAI ?

- FAI macht alles, was ihr Systemadministrator zu tun hat, bevor Sie das erste Mal auf einem neuen Rechner arbeiten können
- Es ist eine Sammlung von Skripten zur vollautomatischen Installation von Debian GNU/Linux (in Zukunft auch Solaris )
- Es kann das ganze Betriebssystem und die Anwendungsprogramme installieren und konfigurieren ■
- Es ist weder ein Cluster Management Tool noch ein Batch System
- Es kann die Installation nicht planen :-), aber ■
- **Plane die Installation und FAI installiert deinen Plan ! :-)**

# Motivation

- Manuelle Installation dauert einige Stunden
- Viele Fragen sind zwischendurch zu beantworten
- Gleiche Daten müssen bei jedem Rechner erneut eingegeben werden
- Manuelle Installationen können nicht parallel durchgeführt werden
- "No simple sysadmin task is fun more than twice"
- Wiederholende Arbeit ist stupide und führt zu Fehlern
- **Ein Installation per Hand skaliert nicht !**

- Automatische Installation dauert nur wenige Minuten
- Garantiert identische Installationen
- Es kann sehr viel Zeit gespart werden !
- Nach Hardwaredefekt ist der Rechner in kürzester Zeit wieder mit identischer Konfiguration einsatzbereit
- Cluster, Serverfarmen und Pools ideal, aber auch heterogene Hardware und Konfigurationen werden unterstützt
- Mit einem Befehl können hunderte Rechner gleichzeitig installiert werden
- Es muß kein Master Image per Hand erzeugt werden

# Wie funktioniert FAI ?



Ein Systemadministrator während der automatischen Installation

# Voraussetzungen

- Rechner mit Netzwerkkarte (install client)
- Nicht nötig: Floppy, CD-ROM, Tastatur, Graphikkarte
- Server mit BOOTP oder DHCP, NFS und TFTP (install server)
- Lokaler Spiegel von Debian (NFS, FTP oder HTTP) ■
- Plattenplatz auf dem Server:

FAI Paket	13 MB	Kernel, Skripte, Konfigurationsdateien
nfsroot	150 MB	erzeugt mit debootstrap
Debian Spiegel	4.2 GB	Debian 3.0 (nur i386)
- Alle Install Clients nutzen die gleichen Verzeichnisse
- **Konstanter Plattenplatz**

# Ablauf einer Installation

- Rechner startet Linux mit NFS-Root, ohne lokale Platte zu benutzen
- Klassen und Variablen definieren
- Kernel Module laden (2.2.20 oder 2.4.18)
- Festplatten partitionieren (mit `setup_harddisks` und `sfdisk`)
- Dateisysteme erzeugen und nach `/tmp/target` mounten (`mke2fs`, `mkswap`, `mkreiserfs`)
- Software Pakete installieren (`apt-get`)
- Betriebssystem und Anwendungen konfigurieren (`shell`, `perl`, `cfengine`, `expect`)

- Protokolldateien lokal und auf Install Server speichern
- Neu installiertes System booten (via TFTP oder von lokaler Platte)
- Installationzeit ohne booten auf Athlon 500 MHz, 320 MB RAM ■

installierte Software	Installationszeit	
	10 MBit LAN	100 MBit LAN
162 MBytes	3 min	2 min
638 MBytes	13 min	10 min
1027 MBytes	23 min	18 min

- **Die Menge der Software bestimmt die Installationzeit.**

# Bootvorgang

- Via Netzwerkkarte mit TFTP und BOOTP oder DHCP
- `make-fai-bootfloppy` erstellt universelle Bootdiskette
- Kernelparameter sind möglich
- FAI Parameter via BOOTP oder DHCP:
  - `$FAI_LOCATION` (T170): Konfigurationsverzeichnis
  - `$FAI_ACTION` (T171): `sysinfo`, `install`
  - `$FAI_FLAGS` (T172): `verbose`, `debug`, `sshd`, `reboot`, `createvt`
- remote Zugriff während der Installation via `ssh`

/etc/bootptab:

.faiglobal:\

:ms=1024:hd=/boot/fai:hn:bs=auto:rp=/usr/lib/fai/nfsroot:

.failocal:\

:tc=.faiglobal:sa=kueppers:ts=kueppers:\

:T170="kueppers:/usr/local/share/fai":T171="sysinfo":\

:sm=255.255.255.0:gw=134.95.9.254:\

:dn=informatik.uni-koeln.de:\

:ds=134.95.9.136,134.95.100.209,134.95.100.208:\

:ys=rubens:yd=informatik4711.YP:\

:nt=time.rrz.uni-koeln.de,time2.rrz.uni-koeln.de:

faiclient99:ha=0x02608c7b40d6:bf=faiclient99:\

:tc=.failocal:T171="install":T172="sshd verbose"

# Die Konfiguration

- hostname, IP Adresse, Partitionierung der Platten, Dateisysteme, Mount Points, Software Pakete, lokale Anpassungen des Betriebssystems und der Anwendungen and applications
- Lokale Anpassungen
  - root Passwort, Accounts, Zeitzone, Tastatur Layout, spezial Kernel , NTP,NIS, inetd, ftp, ssh, lpr, autofs, X11, lilo
  - /etc/fstab, /etc/exports, /etc/hosts.allow
- **Gibt es ein Konzept für all dies ?**

# Das Klassenkonzept

- Für jeden Rechner müssen einige Systemdateien installiert werden
- Für jede Systemdatei existieren mehrere Prototypen
- Die Klassen bestimmen welcher Prototyp benutzt wird
- Alle Teilaufgaben in FAI nutzen auch das Klassenkonzept
- Ein Rechner gehört zu mehreren Klassen
- Klassen werden wie folgt definiert:

**Vordefinierte Klassen:** DEFAULT, hostname und LAST

**In einer Datei:** alle Klassennamen in einer Datei werden genutzt

**Dynamisch über Skripte:** ein Skript gibt z.B. abhängig von der Hardware Klassennamen aus

- Beispiele für Klassen: COMPILER, FAI\_BOOTPART, MBR, DATALESS, NOTEBOOK, DEBIAN\_DEVEL, KERNEL\_SOFT, BOOTP\_SERVER, 4GB, NIS, BEOWULF
- Priorität von niedrig nach hoch

## Variablen

Variablen definieren zusätzlich Parameter

```
.../class/DEFAULT.var:  
FAI_CONSOLEFONT=  
FAI_KEYMAP=us-latin1  
UTC=yes  
time_zone=Europe/Berlin  
rootpw='a3hxVqR5t1t9L'  
moduleslist="3c59x"  
kernelimage=kernel-image-2.2.19-idepci  
printers="kyocera optra hp4si hp hpcolor juenger"
```

```

/usr/local/share/fai/
|-- class
|   |-- ATOMCLIENT.var
|   |-- CS_KOELN.var
|   |-- DEFAULT.mod
|   |-- DEFAULT.var
|   |-- S01alias.sh
|   |-- S03hwdetect.source
|   |-- S05modules.source
|   |-- S24nis.sh
|   |-- S90partitions.pl
|   |-- atoms
|   |-- bigfoot
|   '-- bigfoot.var
|
|-- disk_config
|   |-- 4GB
|   |-- ATOMCLIENT
|   |-- CS_KOELN
|   |-- bigfoot
|   '-- kueppers
|
|-- package_config
|   |-- BOWULF
|   |-- COMPILE
|   |-- DEBIAN_DEVEL
|   |-- DEFAULT
|   |-- GERMAN
|   |-- NFS_SERVER
|   |-- NIS
|   |-- NOTEBOOK
|   '-- pittermaennche
|-- scripts
|   |-- BOOT
|   |-- DEFAULT
|   |-- S01
|   '-- S20
|-- FAI_BOOTPART
|-- LAST
|-- NETWORK
|   |-- S10
|   '-- S40
|-- NOTEBOOK
|-- USB

```

# Platten Konfiguration

- Beispiel /fai/disk\_config/4GB:

```
# <type> <mountpoint> <size> [mount opt] [;extra opt]
disk_config hda
primary / 50 rw,errors=remount-ro ;-c
logical swap 100-200 rw
logical /var 150-200 rw
logical /usr 1500 rw
logical /tmp 100-300 ;-m 1
logical /home 700- rw,nosuid ;-m 0
logical /scratch 0- rw,nosuid ;-m 0 -i 50000
#logical /scratch preserve9 rw,nosuid ;-m 0 -i 50000
```

# Merkmale der Plattenkonfiguration

- Eine Konfigurationsdatei für alle lokalen Platten
- Einfache Angabe von fester oder variabler Partitionsgröße
- Einfache Angabe von Mountpoint und Optionen (`reiserfs`, `ext3`, `nosuid`, `ro`, `-m 0`)
- Daten auf einzelnen Partitionen können erhalten werden (`preserve`)
- Windows Partitionen werden in `/etc/fstab` mit eingetragen
- Automatische Erzeugung von `/etc/fstab`

# Auswahl der Software Pakete

Die Software Installation erledigt ein Perl Skript mit Hilfe von apt-get. Beispiel: /fai/package\_config/BEOWULF:

```
# packages for Beowulf clients
```

```
PACKAGES install
```

```
fping jmon autofs
```

```
rsh-client rsh-server rstat-client rstatd rusers rusersd
```

```
lam-runtime lam3 lam3-dev libpvm3 pvm-dev mpich scalapack-mpich-dev
```

```
dsh update-cluster-hosts update-cluster etherwake
```

```
PACKAGES taskinst
```

```
c-dev
```

- Aktionen wie bei apt-get: install, remove, taskinst
- Abhängigkeiten innerhalb der Pakete werden aufgelöst
- dpkg --get-selections Format auch möglich

# Beispiele für Konfigurations Skripte

```
.../scripts/DEFAULT:
#!/bin/sh
chmod 1777 $target/tmp
chown root:root $target/tmp

# create NIS/NONIS config
fcopy /etc/nsswitch.conf /etc/host.conf
ifclass NONIS && rm -f $target/etc/defaultdomain
if ifclass NIS; then
    echo $YPDOMAIN > $target/etc/defaultdomain
    rm -f $target/etc/yp.conf
    for s in $YPSRVR; do
        echo "ypserver $s" >> $target/etc/yp.conf
    done
fi
# copy default dotfiles for root account
fcopy /root/.bash_profile /root/.bashrc /root/.cshrc
```

# Beispiele für Cfengine Skripte

```
#!/usr/bin/cfengine
control:  actionsequence = ( editfiles )
editfiles:
any::
  { ${target}/etc/passwd
    LocateLineMatching "^root:.*"
    InsertLine      "roott::0:0:root:/root:/usr/bin/tcsh"
    ReplaceAll      "^root::" With "root:${rootpw}:"
    ReplaceAll      "^roott::" With "roott:${rootpw}:"
  }
  { ${target}/etc/default/rcS
    ReplaceAll      "^UTC=.*" With "UTC=${UTC}"
  }
HOME_CLIENT::
  { ${target}/etc/fstab
    HashCommentLinesContaining "/home"
    AppendIfNoSuchLine "${hserver}:/home /home nfs"
  }
```

# Konfiguration mit Prototypen

- für viele Konfigurationsdateien existieren fertige Prototypen
- Unterschiedliche Installationen brauchen unterschiedliche Prototypen
- Die Klassen bestimmen, welcher Prototyp kopiert wird
- Klassenbasiertes kopieren von Dateien mit `fcopy`
- `fcopy` kann rekursiv arbeiten und unterstützt CVS
- Komplette Archive auspacken mit `ftar`

```
/usr/local/share/fai/files
  '-- /etc/nsswitch.conf
    |-- NIS
    '-- NONIS
```

```
DEFAULT/S01: fcopy /etc/nsswitch.conf /etc/host.conf
X11: fcopy /etc/X11/XF86Config /etc/X11/Xserver
```

# Hooks

- Der Installationsprozess besteht aus mehreren Teilaufgaben (z.B. Variablen definieren, Platten partitionieren, Software installieren, Debian mirror mounten, Log Dateien sichern, ...)
- Für jede Standardaufgabe (Task) kann man mehrere Hooks definieren
- Hooks werden klassenbasiert ausgewählt
- Hooks werden vor jedem Task ausgeführt
- Hooks können einen Task resetzen oder erweitern
- Mit hooks kann man die Installation (den Ablauf) sehr einfach an seine Bedürfnisse anpassen

# Informationen über FAI

- Homepage: `www.informatik.uni-koeln.de/fai`
- CVS Baum, eine sehr aktive Mailling Liste, Beispiele der Log Dateien
- Zusätzliche Software während der Installation: `lvm`, `raidtools`, `dump`, `restore`, `ext2resize`, `hdparm`, `parted`, `resize_reiserfs`
- FAI mit Aktion `sysinfo` kann als Rettungssystem benutzt werden. Es mounted alle Partitionen, wenn ein `/etc/fstab` gefunden wird
- Installiert auch Debian auf SUN Sparc (und Solaris auf Sparc)

# Zusammenfassung

- Voll automatische Installation ! Keine manuelle Interaktion
- Garantiert identische, konsistente Installationen
- Einfaches Erstellen der Konfiguration durch Klassen
- Konstanter Plattenplatz auf dem Server
- Alle Konfigurations- und Protokolldateien zentral auf dem Server
- Die schnelle und einfache Reinstallation führt zu einem System mit aktueller Software und wenig Sicherheitsproblemen
- **FAI ist eine skalierende Methode zum Installieren von Debian GNU/Linux**