

# Fully Automatic Installer

Plan your installation, and FAI installs your plan.

Dirk Geschke



Linux User Group Erding

23. März 2011

- 1 Einleitung
- 2 FAI
- 3 Überblick Installation per PXE
- 4 FAI-Server
- 5 Administration
- 6 Praxis

- Verfahren zur **automatischen** Installation vieler Systeme
- **Anpassbare** Systeme
- **Administration** soll auch möglich sein
- **Community**: weit verbreitet im Einsatz
- guter **Support**

**Disk Images** Erstellbar durch dd und Co.

- einfach und schnell
- unflexibel, mitunter viele Images
- hoher Speicherverbrauch durch die Images

**eigene Skripte** selbstentwickeltes System

- Automatisierung
- weniger Speicherverbrauch
- sehr flexibel
- maßgeschneiderte Lösung
- viel Arbeit, da eigenes Produkt
- bei Problemen ist man alleine

**Auto-Installer** Diverse Systeme existieren

**Anaconda / Kickstart** Red Hat, Fedora, CentOS

**AutoYast** SuSE

**Nlite / Unattended** Windows

**Jumpstart** Solaris

**Cobbler/Koan** Red Hat, Fedora, CentOS

**OpenQRM** Cloudansatz - Debian, Ubuntu, CentOS,  
openSuse, ...

**FAI** Debian, Fedora/Red Hat/CentOS, SuSE,  
Solaris, Windows (experimentell)

- Support / Community
- nur für Erstinstallation

# Überblick der Features von FAI

- gute und schnelle **Hilfe** bei Problemen per Mailingliste, IRC
- **automatische** Installation und Administration
- Installation über das Netzwerk per **PXE** oder **BOOT-CD/DVD** bzw. **USB-Stick**
- viele **Interpreter**: `sh`, `perl`, `cfengine`, leicht erweiterbar
- leicht **anpassbar**
- Verwendung von **Klassen**
- Administration **nach** der Installation
- Installation im **Minutenbereich** möglich

## Installation und Administration von

- Clustern
- Arbeitsplatzrechnern
- Schlungssystemen
- VServer
- Webserver

## 1 DHCP liefert

- IP-Adresse
- Netzmaske
- Gateway
- DNS-Server
- TFTP-Server
- TFTP-Datei
- Hostnamen

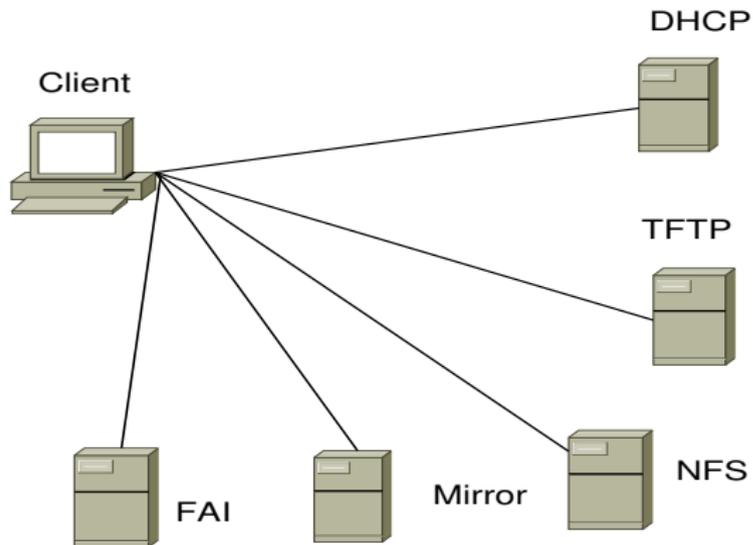
## 2 PXE-Boot

- per TFTP wird NBP (Network Bootstrap Program) geladen
- per TFTP wird dann der Kernel geladen
- Kernel wird mit diversen Parametern gestartet

- ③ Mounten des **Root**-Verzeichnisses per **NFS**
  - Server und Verzeichnis stehen in den Kernelparametern
  - AUFS - Another Union File System wird verwendet, damit können Anpassungen in der RAM-Disk abgespeichert werden
- ④ **Partitionierung** der Festplatte
  - Software-RAID möglich
  - LVM möglich
  - an die Größe anpassbare Partitionen
- ⑤ Anlegen der **Dateisysteme**
  - verschiedene Systeme möglich: `ext2/3/4`, `vfat`, `xf`s, `reiserfs`, ...
  - verschiedene Optionen setzbar, z.B.: `rw`, `noatime`, `nosuid`, `nodev`
  - LVM möglich

# Ein typisches Szenario

- 6 Mounten der Partionen nach `/target/`
- 7 Minimalsystem per `debootstrap` installieren
- 8 **Installation** von Software im `/target/-Verzeichnis` (`chroot`)
- 9 **Konfiguration** und Anpassung der Software/Einstellungen
- 10 Sichern der **Log**-Dateien
- 11 Deaktivieren **PXE**-Installation
- 12 **Reboot**
- 13 **Done.**



- leichte Installation via `fai-quickstart`
- gute **Beispieldateien** existieren
- *relativ* gute Dokumentation auf der Webseite des Projekts  
`http://fai-project.org`
- exportiert Konfigurationsverzeichnis mit Skripten, oft  
`/srf/fai/config`
- exportiert **NFS-Root**-Verzeichnis
- Ideal: lokaler Debian-**Mirror**

# DHCP-Server - dhcpd.conf

```
local-address 172.16.0.1;
deny unknown-clients;
use-host-decl-names on;
subnet 172.16.0.0 netmask 255.255.255.0 {
range 172.16.0.2;
option routers 172.16.0.1;
option domain-name lug-erding.de";
option domain-name-servers 172.16.0.1;
server-name qfix;
next-server qfix;
filename pxelinux.0";
}
host fai-demo {
hardware ethernet 52:54:00:12:34:56;
fixed-address fai-demo; }
```

- gestartet per `inetd.conf`:

```
in.tftpd -s /srv/tftp/fai
```

- Aufbau des Baumes

```
/srv/tftp/fai/  
|- initrd.img-2.6.32-5-amd64  
|- pxelinux.0  
|- pxelinux.cfg  
|  `-- AC100002  
  `-- vmlinuz-2.6.32-5-amd64
```

- Client-IP 172.16.0.2 ist in Hex `AC100002`
- Kann mit dem Programm `fai-chboot` erstellt werden

## Inhalt von AC100002:

```
default fai-generated
```

```
label fai-generated
```

```
kernel vmlinuz-2.6.32-5-amd64
```

```
append initrd=initrd.img-2.6.32-5-amd64 ip=dhcp \
```

```
root=/dev/nfs nfsroot=/srv/fai-squeeze/nfsroot\
```

```
boot=live FAI_FLAGS=verbose,sshd,createvt \
```

```
FAI_ACTION=install
```

- **Verzeichnisbaum**

```
/srv/fai-squeeze/config/
```

```
| - class
```

```
| - debconf
```

```
| - disk_config
```

```
| - files
```

```
| - hooks
```

```
| - package_config
```

```
`- scripts
```

- **Unterverzeichnisse oder Dateien gemäß Klassennamen, z.B.**

```
scripts/FAIBASE
```

**class** Festlegung der Klassen für ein System

**disk\_config** Layout der Festplatte, RAID, LVM

**package\_config** Liste der zu installierenden Pakete

**debconf** Vorkonfiguration von Paketen, z.B. Auswahl von Einstellungen

**scripts** Skripte die pro Klasse ausgeführt werden

**files** Dateien die kopiert werden können, `fcopy`, Skript `postinst` möglich.

**hooks** eigene Skripte zum Schluss der Installation einbaubar

# Klassenbestimmung per `class/50-host-classes`

```
#!/bin/sh
case $HOSTNAME in
    qfix)
        echo "Nicht hier laufen lassen!"
        exit 1;;
    demohost)
        echo "FAIBASE DHCP DEMO" ;;
    gnomehost)
        echo "FAIBASE DHCP DEMO XORG GNOME";;
    *)
        echo "FAIBASE DHCP BASE GERMAN" ;;
esac

(ifclass I386 || ifclass AMD64) && echo GRUB_PC
exit 0
```

- einloggen via `ssh` möglich
- **Verteilung** der Konfiguration per NFS, cvs, svn, HTTP möglich
- `fai -N sofupdate` aktualisiert das System
- `fcopy` mit optionalem `postinst` kopiert und ruft das Skript nur auf, wenn sich die Datei geändert hat
- `ainsl` AppendIfNoSuchLine-Programm zur Administration
- `cfengine` zur Bearbeitung von Konfigurationsdateien, Erstellen von Links, etc.
- Shell-, Cfengine- und Perlskripte, Aufruf beliebiger installierter Programme

# Praxis

- effektives Werkzeug zur Installation vieler Systeme
- Verwaltung ist ebenfalls möglich
- leicht anpassbar
- sehr flexibel
- gute Community
- hohe Anfangsschwierigkeiten
- ein fürchterlicher Verhaù an Skripten

Fine!