Das Installationstool FAI (Fully Automatic Installation)

Thomas Lange
Institut für Informatik
Universität zu Köln

E-mail: lange@informatik.uni-koeln.de

www.informatik.uni-koeln.de/ls_juenger/lange.html

Kolloquium Praxis der Datenverarbeitung
WS 2000/2001
22. November 2000

Überblick

- Was ist FAI ?
- Motivation
- Voraussetzungen
- Ablauf einer Installation
- Klassenkonzept
- Konfiguration
- Ausblick

Was ist FAI ?

- FAI macht alles, was ihr Systemadministrator zu tun hat, bevor sie sich das erste mal auf einem Rechner einloggen können

- Es ist eine Sammlung von Skripten zur vollautomatischen Installation von Debian GNU/Linux

- Es kann das ganze Betriebssystem und die Anwendungsprogramme installieren und konfigurieren

- Was ist FAI nicht:
  - Batch Queue System
  - Tool für automatische Parallelisierung von Programmen
  - Resource Management Tool
  - Netzwerk Management Tool
  - Performance Analyse Tool
  - Cluster Monitoring Tool
  - Es kann aber all diese Dinge installieren und konfigurieren

Eigenschaften von FAI

- Voll automatische Installation ! Keine Interaktion zwischendurch notwendig
- Kann auch bei defekter Festplatte zum Backup und zur Fehlersuche genutzt werden (Rescue System)
- Konfiguration erstellen für einen neuen Rechner:
  - bei $bekannter$ Hardware: 1 Minute
  - bei $neuer$ Hardware: mehrere Minuten bis zu einer Stunde (grob abgeschätzt)
- Konstanter Plattenplatz auf dem Server
- Zentrale Speicherung aller Konfigurations- und Protokolldateien aller Rechner auf dem Server
- Garantiert identische Installationen
- **Deshalb: FAI ist eine skalierende Methode zum Installieren von Debian GNU/Linux**

Motivation

- Wie läuft bei ihnen eine Debian GNU/Linux Installation ab ?
  - Boot- und Root-Diskette erstellen oder CD-ROM nutzen, Booten des Rechners
  - Tastatur, Sprache auswählen, Zeitzone, Datum/Uhrzeit, ... einstellen
  - Festplatten partitionieren, Dateisysteme erstellen und Mountpoints definieren (ohne Optionen !)
  - Treiber für Netzwerkkarte installieren
  - Netzwerk konfigurieren: Rechnername, IP-Adresse, Netzmaske, Gateway
  - Rechner booten
  - Paket Tool konfigurieren (dselect)
  - Softwarepakete auswählen, installieren und konfigurieren
  - Root Paßwort setzen, Accounts einrichten, Drucker definieren, X11 konfigurieren
  - Dienste ein- oder ausschalten und konfigurieren (ftp, sendmail, pop, imap, sshd...)

Der Zeitaufwand für eine Installation beträgt einige Stunden pro Rechner, mit viel Wartezeit zwischendurch. Man kann nicht alle Informationen am Anfang eingeben damit der Rest von alleine durchläuft.

Viele Dinge sind bei allen Rechnern gleich (z.B Gateway, Zeitzone, Softwareauswahl) müssen jedoch bei jedem Rechner und bei jeder Installation neu eingegeben werden

- Sich wiederholende Arbeit ist langweilig und führt zu Unaufmerksamkeit und Fehlern
- Automatische Installation garantiert identische Installationen
- Es kann sehr viel Zeit gespart werden !
- Auch die Installation eines einzelnen Rechners lohnt sich
- Nach einem Hardwaredefekt, kann der Rechner in kürzester Zeit wieder erneut installiert werden
- Erste Erfahrungen kamen von Jumpstart für Sun Solaris
- Cluster sind ideal, da alle Rechner die gleiche Hardware besitzen (Informatik Cluster 1999: 16 Stück dual PII 400Mhz je 265 MB, vernetzt über Switch mit 100 Mbit, ohne Monitor und Tastatur)

Voraussetzungen

- Ein Rechner mit Netzwerkkarte, der mit Debian GNU/Linux installiert werden soll
- Ein vorhandener Server mit BOOTP oder DHCP, NFS und TFTP Diensten (muß kein Linux sein)
- Ein lokaler Spiegel von Debian (NFS, FTP oder HTTP Zugriff) wird empfohlen
- Plattenplatz auf dem Server:

FAI Paket	10 MB	Skripte und lokale Konfiguration
NFS-Root Dateisystem	90 MB	base2_2.tgz ausgepackt und weiteres
Debian Mirror	2GB	Debian 2.2 (potato, nur i386)

  Da alle Rechner auf diese Verzeichnisse nur lesend zugreifen, können sie von beliebig vielen Rechner geteilt werden. Konstanter Plattenplatz !

Ablauf einer Installation

Kurzfassung:

- Rechner booten
- Festplatten partitionieren
- Software installieren
- Anpassungen machen
- neues System booten

Ausführlich:

1. MAC-Adresse feststellen, IP-Adresse und Rechnername festlegen
2. Konfiguration erstellen
3. Rechner bootet, fragt nach seiner IP-Adresse und holt den Kernel (BOOTP/DHCP, TFTP)
4. Startet ein voll funktionsfähiges Debian GNU/Linux ohne die lokalen Platten zu benutzen, nur die Ramdisk ist schreibbar. Mountet dabei NFS-Root Dateisystem.

5. weitere Parameter via BOOTP/DHCP holen (server,FAI_ACTION)
6. FAI Konfiguration `/fai` mounten
7. Klassen definieren
8. Kernel Module laden
9. Platte partitionieren, Dateisysteme erstellen
10. Grundsystem installieren (notwendige Pakete)
11. Ausgewählte Softwarepakete installieren
12. Konfiguration anpassen (cfegine, perl, shell oder expect Skripte)
13. Protokolldateien lokal und auf dem Server speichern
14. reboot des neuen Systems

Installationszeit für Schritte 6 bis 13 auf Pentium II 400 Mhz, 128 MB RAM, 10Mbit LAN,

- 83 MB Software : 2 Minuten
- 470 MB Software : 10 Minuten
- 3.3 GB auf defekte Blöcke testen: ca. 6 Minuten

Die Installationszeit bleibt bei einem Client und 100Mbit LAN gleich. **Die Installationszeit wird durch die Menge der Software bestimmt.**

Die Konfiguration

- Wie heisst der Rechner ? Welche IP-Adresse ?
- Wie sollen die Platten partitioniert werden ?
- Wo und wie werden die Partitionen gemountet ?
- Welche Software soll wie installiert werden ?
- Wie soll das System den lokalen Gegebenheiten angepasst werden ?
- **Ist für all das ein Konzept vorhanden ?**

Die abschließenden Skripte machen alles, was ein Systemadministrator nach einer ersten Installation per Hand ändert. Beispiele:

- Root Passwort setzen
- ftp, NTP Dienst einschalten und konfigurieren
- NIS aufsetzen, Accounts einrichten
- X11 konfigurieren
- `/etc/fstab, /etc/exports` editieren
- angepaßten Kernel installieren

Die Konfiguration liegt in Verzeichnissen in: `/usr/local/share/fai`

fai_config/ Globale Parameter ohne Klassen

class/ Definitionen von Klassen und Parameter und Module laden

disk_config/ Plattenkonfigurationen (Partitionen und Mountpoints)

package_config/ Softwarepaketkonfigurationen

scripts/ Skripte für lokale Anpassungen (cfengine, perl, shell, expect)

files/ Vorbereitete Dateien, die in das System hineinkopiert werden (eigene Kernel, dot files, hosts.allow, printcap)

BOOTP Daten

bootpc.log:

```
SERVER='134.95.9.143'
IPADDR='134.95.9.200'
BOOTFILE='/boot/fai/faiserver'
NETMASK='255.255.255.0'
NETWORK='134.95.9.0'
BROADCAST='134.95.9.255'
GATEWAYS_1='134.95.9.254'
GATEWAYS='134.95.9.254'
ROOT_PATH='/usr/lib/fai/nfsroot'
DNSSRVS_1='134.95.9.136'
DNSSRVS_2='134.95.100.209'
DNSSRVS_3='134.95.100.208'
DNSSRVS='134.95.9.136 134.95.100.209 134.95.100.208'
DOMAIN='informatik.uni-koeln.de'
SEARCH='informatik.uni-koeln.de uni-koeln.de'
YPSRVR='134.95.9.10'
YPDOMAIN='informatik4711.YP'
TIMESRVS_1='134.95.9.10'
TIMESRVS='134.95.9.10'
HOSTNAME='faiserver'
T170='kueppers:/usr/local/share/fai'
T171='sysinfo'
T171='install'
T172='sshd verbose debug'
```

Festplattenkonfiguration

Im Verzeichnis `disk_config/` gibt es beispielsweise die Datei `4GB`:

```
# disk configuration for one disk with up to 4GB disk space
# <type> <mountpoint> <size in mb> [mount options]     [;extra options]

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
```

Features der Plattenkonfiguration

- Einfache Angabe von Größe, Mountpoint und Optionen
- Partitionsgröße fest oder variabel
- Auf einzelnen Partitionen können vorhandene Daten bewahrt werden (preserveN, kein mkfs ausführen)
- Optionen für mkfs (z.B. reserved blocks, check for bad blocks, $FS\ Typ$)
- Optionen für mount Befehl (z.B. nosuid, ro)
- Windows Partitionen können in `/etc/fstab` mit eingetragen werden
- Automatische Erzeugung von `/etc/fstab` Einträgen
- Es wird nur eine Konfigurationsdatei benutzt

Nach der Installation mit 4GB:

```
Filesystem 1k-blocks   Used Available Use% Mounted on
/dev/hda1      54447  12446     39190  24% /
/dev/hda6     171113  23096    139182  14% /var
/dev/hda7    1517948 442304    998536  31% /usr
/dev/hda8     132221    162    130694   0% /tmp
/dev/hda9    1351948     20   1351928   0% /home
/dev/hda10    793084     20    785112   0% /scratch
```

Paket Konfiguration

Beispiel für die Klasse `COMPILE`:

```
# COMPILE: packages for developing software
PACKAGES install
task-c-dev task-c++-dev
bin86 binutils m4
libc6-dev libg++2.8.1.3 libstdc++2.10-dev
libstdc++2.10 g77 byacc cvs
```

- Aktionen: install, remove
- Es müssen keine Versionsnummern angegeben werden (im Beispiel gehören die Zahlen zum Paketnamen)
- Abhängigkeiten werden aufgelöst, d.h. Pakete, die von anderen gebraucht werden, werden automatisch mit installiert
- in Zukunft auch Format von `dpkg --get-selections` möglich

Das Klassenkonzept

Die Konfiguration wird aus Bausteinen (Klassen) zusammen gesetzt. Eine Klasse ist definiert oder nicht, sie hat keinen besonderen Wert. Konfigurationsdateien, die den Namen einer definierten Klasse haben, werden bei der Konfiguration automatisch mitbenutzt. Dadurch kann die Konfiguration sehr leicht erweitert werden, ohne die Skripte zu ändern.

- Klassennamen bestehen aus: [0-9A-Z_]* (Ausnahme hostname;

- Alle Klassen werden über Skripte und Dateien in `class/` definiert

- Klassen können auf verschiedene Arten für einen Rechner definiert werden:

  Vordefinierte Klassen: hostname, ALL und LAST
  In einer Datei: alle Klassennamen in einer Datei werden genutzt
  Dynamisch über Skripte: ein Skript gibt Klassennamen aus

Beispiele:

S01alias.sh:

```
#! /bin/sh
# all roy's are using classes in roy.classes

case $HOSTNAME in

    roy??)
        cat roy.classes
        ;;

esac
```

S24nis.sh:

```
#! /bin/sh
# add NIS if YPDOMAIN is defined

if [ -n "$YPDOMAIN" ];then
   echo NIS
   echo $YPDOMAIN | tr '.a-z-' '_A-Z_'
else
   echo NONIS
fi
```

S07disk.pl:

```
#! /usr/bin/perl

# define classes for different disk configurations

# global variables:
# $numdisks            # number of disks
# %disksize{$device}   # size for each device in Mb
# $sum_disk_size       # sum of all disksizes in Mb

use Debian::Fai;

read_disk_info();

# rules for classes
#-----------------------------------------------
# two SCSI disks 2-5 GB
($numdisks == 2) and
    disksize(sda,2000,5000) and
    disksize(sdb,2000,5000) and
    class("SD_2_5GB");

# one disk 1-4 GB, IDE or SCSI
($numdisks == 1) and
    testsize($sum_disk_size,1000,4000) and
    class("4GB");
```

Beispiel für Konfigurationsskripte

```
ALL:

#! /bin/sh
# dump keytable
dumpkeys | gzip -9f \
     >$target/etc/console-tools/default.kmap.gz

# if $kernelimage if not a debfile,
# then it's the kernel version
# create dummy link, so package
#  kernel-image-* makes new correcxt link
ln -s /boot/vmlinuz-nolink $target/vmlinuz

$k=$files/packages/$kernelimage
if [ -f $k ]; then
   /usr/bin/yes 'n' | /usr/bin/dpkg --root=$target -i $k
else
   # default kernel
   yes 'n' | \
   chroot $target apt-get install kernel-image-$kernelimage
fi
-------------------------------------------------------
RESOLV:

#! /bin/sh
# create resolv.conf

[ -s /tmp/etc/resolv.conf ] && \
   cp -p /tmp/etc/resolv.conf $FAI_ROOT/etc

for class in $classes; do
    resolv=$FAI_FILES/etc/resolv.conf/$class
    [ -f $resolv ] && cp -p $resolv $FAI_ROOT/etc/resolv.conf
done
```

Beispiel für cfengine

```
#! /usr/bin/cfengine

control:
   actionsequence = ( editfiles )

editfiles:
 { ${target}/etc/init.d/sysklogd
   ReplaceAll "^SYSLOGD=.*" With
       "SYSLOGD=${dblquote}-m 240${dblquote}"
 }

 { ${target}/etc/passwd
   LocateLineMatching "^root:.*"
   InsertLine   "rootт::0:0:root:/root:/usr/bin/tcsh"
   ReplaceAll   "^root::" With "root:${rootpw}:"
   ReplaceAll   "^rootт::" With "rootт:${rootpw}:"
 }

 { ${target}/etc/timezone
   AutoCreate
   EmptyEntireFilePlease
   Append "${time_zone}"
 }
```

Mini Tutorial

Wie installiere ich einen neuen Rechner ?

- MAC-Adresse feststellen und einem Rechnernamen zuordnen, IP-Adresse festlegen
- BIOS umstellen, damit von Netzwerkkarte gebootet wird oder Bootdiskette erstellen
- `bootptab` Eintrag mit `T171="install"`
- `ln -s installimage /boot/fai/<hostname>`
- Rechner zur Netgroup *@faiclients* hinzufügen (`/etc/netgroup` oder NIS)
- Root Passwort definieren:
  `rootpw='87hdRgt34t5C'`
  (z.B. in `class/S98variables.source`)
- Zeitzone: `time_zone=Europe/Berlin`
- Lilo Parameter:
  `liloappend='append="mem=320M ramdisk=20000"'`
- Kernelimage: default oder mit make-kpkg ein Paket erzeugen und angeben
  `kernelimage=kernel-image-2.2.15_kueppers1_i386.deb`
- Plattenlayout bestimmen
- Klassen definieren (z.B wie in class/faiserver)
- Rechner booten (Er installiert sich)
- rcS.log, software.log ,... ansehen
- Rechner von lokaler Festplatte booten

Technische Eigenschaften von FAI

- Homepage: `www.informatik.uni-koeln.de/fai`
- Maillingliste, CVS repository
- Verfügbar als (offizielles) Debian Paket
- Keine initial ramdisk notwendig, dadurch wenig RAM nötig (mit 8MB getestet)
- Läuft auch auf 386 CPU
- Ein einfaches Skript erzeugt die Bootfloppy
- Lilo Parameter für den FAI Kernel möglich
- Installationskernel 2.2.17 mit Modulen wird als Paket installiert (make-kpkg)
- Zusätzlich während Installation verfügbar sind: lvm, raidtools, dump, restore, ext2resize, strace
- Remote Zugriff über ssh (reboot möglich) und zwei weitere Konsolen während der Installation
- Wählbares Tastatur Layout während der Installation (z.B für deutsche Tastatur)
- Zugriff auf Pakete via NFS, FTP oder HTTP
- Verschiedene Aktionen möglich (install, sysinfo, showclasses, $backup$)
- Alle Protokolldateien werden auch auf dem Server gespeichert
- FAI mit Aktion `sysinfo` kann als rescue System benutzt werden. Mountet alle lokalen Partitionen automatisch, findet /etc/fstab
- `FAI_FLAGS` für verbose, debug, reboot, sshd

Ausblick

- Tutorial und Manual ist in Arbeit
- Nächste Version (1.4.0) in ca. 1-2 Tagen, danach kommt erste offizielle Debian Version
- Ganz neues Programm zur Plattenkonfiguration, das auch andere Dateisystemtypen unterstützt
- Test im harten Einsatz auf großem Cluster (>30 Knoten) oder in inhomogener Umgebung wäre schön. Bei Interesse bitte bei mir melden !