

Automatische Installationen und System Updates mit FAI

Übersicht, Funktionalität, Möglichkeiten

Henning Sprang

Freier EDV Berater, Autor und Trainer

Newthinking Store Berlin 2009-02-03

Der Redner - Henning Sprang

- FAI Team-Mitglied
- IT Berater, Autor, Trainer
- Interessen: System-Management, Virtualisierung, Java, QA
- Freie Software Enthusiast seit dem ersten Kontakt ca. 1996

Fragen?

Einfach fragen!

Vorschau

- 1 Über System-Installationen
- 2 Über FAI
- 3 Funktionen
- 4 Details der Nutzung
- 5 Ausblick

Einige Fragen...

Was tun Sie...

- Wenn Ihr wichtigster Server gerade abgebrannt ist?
- Wenn ein Entwickler ein neues Test-System benötigt?
 - Mit GENAU der gleichen Konfiguration wie das Produktiv-System!
- Wenn Sie ein neues Notebook gekauft haben?
- Wenn Sie die Konfiguration eines Application Server Clusters ändern müssen?

System Installationen - Anforderungen

- Systeme nach einem Disaster wiederherstellen
- Systeme exakt reproduzieren und multiplizieren
- Mehrere Systeme gleichzeitig und schnell installieren
- Nach der Installation müssen Änderungen verwaltet werden!

Installation - Wie geht man vor?

- Manuelle Installation hat Nachteile:
 - Nicht gut reproduzierbar - Menschen machen Fehler
 - Ineffizient - mehrere Systeme installieren dauert...
- Lösung: Automatisierung
- Nebeneffekt: Die Konfiguration ist dokumentiert

Disk Images

.img?

- Vorlagen erzeugen von funktionierenden Installationen
- Bei Bedarf mit dd aufspielen und anpassen
- Vorteile:
 - Niedrige Lernschwelle
 - Schnell und einfach implementiert – cp/rsync/tar/dd
- Nachteile:
 - Unflexibel – bei der kleinsten Änderung neues Image machen
 - Immer noch Handarbeit nötig beim Duplizieren von Konfigurationen
 - Speicherkosten linear zur Anzahl der unterschiedlichen Konfigurationen

Installation mit Shell Skripten

#!/bin/bash

- Manuelle Arbeit wird durch Shell Skripte ersetzt
- Vorteile:
 - Wesentlich geringerer Speicherbedarf als Images
 - Sehr flexibel (alles, was man skripten kann)
 - Maßgeschneidert
- Nachteile:
 - Plötzlich hat man ein neues SW-Entwicklungsprojekt am Bein
 - Man muss alle Probleme selbst lösen

Nutzung verfügbarer Auto-Installer

Lösungen sind verfügbar. . .

Die wichtigsten Projekte und Ziel-Distributionen:

- Anaconda und Kickstart: Fedora-basierte
- Autoyast: SuSE-basierte
- Nlite/Unattend: Windows in unterschiedlichen Varianten
- FAI: Debian-basierte, Fedora-basierte, SuSE-basierte, Windows und Solaris teilweise
- Andere: Solaris Jumpstart, RedHat Cobbler/Koan

Warum FAI?

- Flexibel und leicht zu erweitern
- Einfache, mächtige Architektur – alles ist ein Shellskript
- Support durch erfahrene “Installateure” (frei+kommerz.)
- Diverse Client- und Server- Distributionen
- Mehrere Installations-Arten plus Updates
- Kann für echte Hardware und Virtualisierung genutzt werden

Geschichte

- Start 1999 von Thomas Lange an der Universität zu Koeln
- Grundlegende Idee: Strukturierte und geplante Installation
- „Plane Deine Installation und FAI installiert Deinen Plan“
- Teil der Debian Distribution
- Heute ca. 10 Entwickler – eine kleine aber feine Community
- Seit 2005 “Softupdates” – System-Management!

Wer nutzt FAI wofür?

- EDF nutzt FAI (mit GOSA) für einige Forschungs-Cluster
- LiMux in München: installiert/aktualisiert Clients und Server
- Mehrere top 500 High Performance Cluster
- Kleine Heim-Netzwerke ab 2 Systeme
- GRML live CD wird mit FAI gebaut

Übersicht I

FAI Klassen

- Eine Klasse definiert Systemeigenschaften und Aktionen
- Klassen-Zuweisung per Script (+ Textdatei, DB, ...)
- Ein System kann zu beliebig kombinierten Klassen gehören

Übersicht II

- Server-Distribution:
 - Debian-basierte (Depends: Perl, NFS, TFTP, debootstrap)
- Ziel-Distributionen:
 - Debian, Redhat, Ubuntu, Suse, Mandriva, Solaris
- mehrere Installationsarten
- Integrierte Versionierung mit Subversion, CVS, git
- Softupdate für Aktualisierungen

Installations-Typen

- Netzinstallation mit zentralem Installations-Server
 - Client/Server Architektur
- Installation in ein Verzeichnis (chroots, Vserver, . . .)
- CD/USB für Umgebungen ohne Netzwerk
- Mit grml-live: Live-CD's erstellen!

Die wichtigsten *tasks*/Schritte der Installation:

- *defclass*: Klassenzuweisung für das Zielsystem
- *partition*: Partitionierung
- *extrbase*: Minimales Basis-Image auspacken
- *debconf*: Debconf preseedings anwenden
- *instsoft*: Installation der Software-Pakete
- *configure*: Konfigurations-Skripte ausführen
- *savelog*: Logfiles sichern

Vorüberlegungen – Installation

- Entscheidung für den richtigen Installations-Typen (Netz/CD/dirinstall)
- Plane Deine Installation
 - Nutzungs-Szenarios
 - Netzwerk und Betriebsumgebung
 - Software-Pakete
 - Zusätzliche Dateien und Anpassungen
- Mirrors der APT-Repositories

Vorüberlegungen - Updates

- Wie und wann sollen Patches erfolgen?
- Testing – welchen Effekt wird ein Patch haben?
- Mirrors auch für Security Updates von Paketen?
- Automatisch(zeitbasiert, beim Starten) or manuell gestartet?

FAI Setup und Konfiguration I

- Installation auf Debian/Ubuntu:
 - *apt-get install fai-quickstart*
- Anpassen der Konfiguration in /etc/fai:
 - fai.conf: LOGUSER=fai, LOGPROTO=ssh for Logging via ssh
 - apt/sources.list: Lokalen Mirror nutzen wenn verfügbar

FAI Setup und Konfiguration II

- FAI Server ist eingerichtet – NFSRoot anlegen:
 - *fai-setup* / *make-fai-nfsroot*
- Für PXE-Boot: *fai-chboot* für Boot-Konfiguration
- Ohne PXE / bei CD-Installation: *fai-cd*

Infrastruktur-Dienste aufsetzen

Für Netzinstallationen:

- DNS Einträge für Server und Clients
- DHCP Konfiguration - Host/IP/MAC wie üblich
- FAI-spezifisch:

```
authoritative; # IP_PNP_DHCP
option root-path "/usr/lib/fai/nfsroot ..."
server-name "faiserver"; # boot-server
next-server 172.20.2.64; # tftp server: kernel
filename "pxelinux.0";
```

System-Konfigurationen anlegen

Konfigurationen sind im FAI *Configspace* abgelegt

- Einfache Textdateien und Skripte
- Bildet Anforderungen aus dem Installations-Plan ab
- Beispiele in `/usr/share/doc/fai/examples/simple`
- Default Speicherort: `/srv/fai/config`

Der FAI Configspace

Inhalte des Configspace:

```
class
disk_config
basefiles
debconf
package_config
scripts
files
hooks
```


Configspace anpassen - class

- *class* enthält Klassenzuweisungen und Variablen-Definitionen.
- Am einfachsten: Klassen anhand Hostnamen zuweisen
- Beispiel-Klassen: FAISERVER, GNOME, DEMO, XORG
- Jedes beliebige Skript kann genutzt werden – auch DB
- Z.B.: Prüfen auf bestimmte Hardware, MAC, IP, ...

Configspace anpassen – basefiles

- Minimale Basis-Images für nicht-Debian-Distributionen
- Bei *extrbase* wird geprüft, ob hier ein passendes Image liegt
- Zuordnung nach Klassen
- Man könnte auch nur dieses Image nutzen. . .

Configspace anpassen – disk_config

Festplatten-Konfiguration und Mountpoints definieren wie benötigt

```
# <type> <mountpoint> <size> <fs type> <mount options> <misc>

disk_config disk1      preserve_reinstall:5,8,9

primary /      250      ext3  rw,errors=remount-ro
logical swap  500-1500  ext3  rw
logical /var   500-1000  ext3  rw createopts="-m15" tuneopts=""
logical /tmp  500-1000  ext3  rw createopts="-m1"  tuneopts=""
logical /usr  500-3G   ext3  rw tuneopts="-c0 -i0"
logical /home 100-     ext3  rw,nosuid  createopts="-m1"
logical /scra 1G-50%  ext3  rw,nosuid  createopts="-m0" tuneop
```

Configspace anpassen - debconf

- Voreinstellungen für Paket-Installations-Skripte
- Nur für dpkg-Distributionen
- Funktioniert wie beim Debian Installer

Configspace anpassen - package_config

- Inhalte: Dateien, benannt nach Klassen
- Zweck: zu installierende Pakete definieren
- Unterstützt mehrere Installationsarten:
 - install (apt-get)
 - aptitude
 - taskinst (Debian tasks=Paketgruppen)
 - urpmi (mandriva)
 - yum (Fedora)
 - y2i (SuSE y2pms)
 - yast (SuSE yast -i)

Configspace anpassen - package_config II

Example package_config/DEMO aus den Simple Examples:

```
PACKAGES aptitude  
fortune-mod fortunes  
rstat-client #rstatd  
rusers rusersd
```

```
# only when also class XORG is defined  
PACKAGES aptitude XORG  
bb frozen-bubble xpenguins
```

Configspace anpassen - scripts

- *scripts* enthält Skripte die nach der Paketinstallation laufen
- Default: Shell- Perl- und Cfengine-Skripte
- Andere Interpreter in NFSRoot installieren ist einfach!
- Namens-Schema:
 <CLASSNAME>/<NUMBER>-<SCRIPTNAME>
- Die Zahl definiert die Ausführungs-Reihenfolge
- SCRIPTNAME beliebig - nur für die Lesbarkeit

scripts Beispiel:

```
.  
|-- AMD64  
|   |-- 99-discover-bug  
|-- DEMO  
|   |-- 10-misc  
|   |-- 30-demo  
|-- FAIBASE  
|   |-- 10-misc  
|   |-- 30-interface  
|   |-- 40-misc  
|-- FAISERVER  
|   |-- 10-conffiles  
|   |-- 20-copy-mirror  
|-- GRUB  
|   |-- 10-setup  
|-- LAST  
|   |-- 50-misc
```


Configspace anpassen - files

- Dateisystem-Hierarchie, angefangen bei /
- Zur Nutzung mit fcopy/ftar
- Klassenbasiertes Kopieren/Entpacken
- fcopy Kopiert einzelne Dateien oder rekursiv aus /
- ftar entpackt ein Archiv...

Configspace anpassen - files II

- Zieldatei=Verzeichnis
- Tatsächlich auf Zielsystem kopierte Datei=CLASSNAME

```
.
|-- etc
|  |-- X11
|     |-- xorg.conf
|         |-- ATI
|             |-- NVIDIA
|-- apache2
|     |-- conf.d
|         |-- debian-mirror.conf
|             |--FAISERVER
|-- fai
|     |-- fai.conf
|         |-- FAISERVER
|             |-- FAI_SOFTUPDATE_CLIENT
```

Configspace anpassen - hooks

- Namens-Schema für Hooks: `.<CLASSNAME>[.source]`
- Werden vor dem entsprechenden Task ausgeführt
- Optionales `skip_task`: Der Task selbst wird nicht ausgeführt
- Beispiele:

```
partition.XENU  
instsoft.FAIBASE  
savelog.LAST.source
```

Installation durchführen

Je nach gewählter Installations-Art:

- Booten per PXE/Boot-CD/install-CD/USB Stick
 - PXE-config setzen mit `fai-chboot`
- `fai dirinstall <TARGETDIR>` in ein gemountetes Blockdevice
- `dirinstall` aufrufen aus `xen-tools`, `Ganeti`, ...
- System mit `fai-cd/fai-usb` starten
- `grml-live` ausführen

System (neu) starten

- System ist in 3-20 Minuten installiert(je nach Paketmenge)
- Neustart mit Produktions-Konfiguration
- Funktions-Tests

Ausblick

- Beispiel-Konfigurationen und Hilfsskripte für andere Distributionen: *fai-distributions*
- GOSA als LDAP und FAI GUI
- Automatische Tests der installierten Systeme
 - hooks/scripts könnten Konfigurationsdateien prüfen
 - Crucible Test Framework /eher für Entwicklung)
- Kein LDAP? Management-Tool/lightweight GUI ohne GOSA!

Weitere Informationen

- WWW:
 - <http://www.informatik.uni-koeln.de/fai/>
 - <http://faiwiki.informatik.uni-koeln.de/>
 - <http://www.infrastructures.org/>
- Email und Chat:
 - IRC-Channel #fai im OFTC-Network
 - linux-fai-users and linux-fai-devel Mailing Listen

Fazit

- FAI ist ein mächtiges Werkzeug für
 - Automatische Installation
 - System-Change-Management
 - Live-CD's bauen mit grml-live
- Strukturiertes Vorgehen ist Voraussetzung
- Einarbeitung kann fordernd sein
- Der Weg lohnt sich aber. . .

Kontakt

Kontakt zum Redner:

- henning@sprang.de
- <http://www.sprang.de>

Fragen?

Fragen?