FAI - a tool for the fully automatic installation of Debian GNU/Linux

Thomas Lange
Institute of Computer Science
University of Cologne
lange@informatik.uni-koeln.de

Linux Kongress, november 29, 2001

Introduction

- Administration is the sum of installation, configuration and daily maintenance

- Installation is the basic administration task

- Bad system administration makes computers unusable

- Much time has to be spend in cluster installation and management

- Today, installation of masses of computers is a major challenge

- A scalable system administration procedure is a must

The real world

- More and more employees are working with a computer

- Commodity hardware is cheap and fast (and often faulty)

- Clusters become hip, but are often run without sysadmin staff

- Beowulf example: 26×1GHz Athlon, 256MB RAM, 20GB disk, Fast Ethernet switch, cables, shelf and DAT-drive bought for 20.000 Euro at the end of 2001 by an institute of physical chemistry

- Know-how is missing and few time is spend in planing a cluster

- Without a sysadmin, no time is spend in maintaining a running cluster (security patches, new OS version)
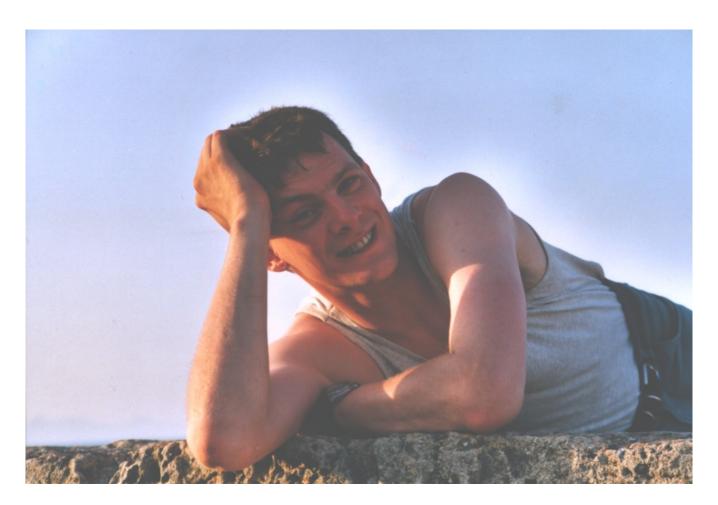
What is FAI ?

- FAI does everything a sysadmin has to do (you!), before users can log in a brand new computer for the first time

- It is a collection of scripts and configuration files

- It installs the whole operating system and all applications

- It is an easy to handle installation tool for Beowulf clusters ▮

- It's neither a cluster management tool nor a job scheduling system

- It can't plan your installation :–(

- But: **Plan your installation and FAI installs your plan ! :–)**

Motivation

- A manual installation takes long time (several hours)

- You have to answers many questions

- Equal configuration data must be entered for every new computer again and again

- Repeating work is boring and leads to errors

- You can't do multiple installations concurrently

- **A manual installation does not scale !**

- An automatic installation only takes several minutes

- It guarantees multiple identical configurations

- It allows quick reinstallation after replacement of defective hardware with same configuration as before

- You can install hundreds of machines with one command

- You can save much time ! (And do other nice things)

- I'm too lazy for manual installations

- Watching a fully automatic installation is fun

How does FAI work ?

A system administrator during a fully automatic installation

Requirements

- A computer with network interface card called install client

- An install server with BOOTP or DHCP, NFS and TFTP daemon

- Access to a local Debian mirror via NFS, FTP or HTTP

- Disk space on the install server:

FAI package	10 MB	kernel, scripts and configuration data
Client nfsroot	100 MB	made of the Debian base system
Debian mirror	2.5 GB	a partial mirror for i386 only

- All install client share the same nfsroot

- **Constant disk space**

Sequence of an installation

- Install client boots linux using the nfsroot, without using the local hard disks

- Define classes and variables and load kernel modules

- Partition local hard disk (using `setup_harddisks` and `sfdisk`)

- Create file systems (`mke2fs, mkswap, mkreiserfs`)

- Debian command `apt-get` installs software and resolves package dependencies

- Configure operating systems and applications

- Save log files to the local disk and to the install server

- Boot the newly installed system

- Time for one installation without booting of a Dual Pentium II 400 MHz, 128 MB RAM, 10Mbit LAN

  - 90 MB software: 2 minutes
  - 520 MB software: 11 minutes
  - test 4 GB for bad blocks: 6 minutes

- The time for installation remains unchanged for one client and 100Mbit LAN.

- **The amount of software determines the installation time.**

Booting

- Via a bootable network card using TFTP and BOOTP, DHCP or PXE method

- Via a common boot floppy which is created by the command `make-fai-bootfloppy`

- Kernel parameters can be specified

- FAI parameters are passed via BOOTP or DHCP:

  - `$FAI_LOCATION (T170)`: location of the configuration
  - `$FAI_ACTION (T171)`: sysinfo, install, backup, …
  - `$FAI_FLAGS (T172)`: verbose, debug, sshd, createvt, reboot

- Remote access during the 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=rubens:\
  :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"
```

The Configuration

- hostname, IP-address, hard disk partition layout, file systems, mount points, software to install, local configuration of operating system and applications

- local customization

  - root password, accounts, timezone, keyboard layout, special kernel, NTP,NIS, inetd, ftp, ssh, lpr, autofs, X11, lilo
  - `/etc/fstab, /etc/exports, /etc/hosts.allow`

- It is the sum of: classes, variables, scripts, templates

- **Do you have a concept for all this ?**

The class concept

- For each host many system files must be installed

- For each system file several templates are available

- A class specifies which template to use

- An install client belongs to several classes

- The configuration is composed of classes

- The selection of configuration files for the disk partitioning, the software installation and for all configuration scripts also uses classes

- Classes can be defined as follows:

  predefined classes: DEFAULT, LAST and the hostname
  stored in a file: Classes are listed in a text file
  dynamically generated: Scripts can define classes depending on
  detected hardware and write them to stdout

- Adding a class easily expands the configuration for a host

- Examples of classes: COMPILER, FAI_BOOTPART, MBR, DATALESS, NOTEBOOK, DEBIAN_DEVEL, KERNEL_SOFT, BOOTP_SERVER, 4GB, NIS, BEOWULF

Variables

Variables define additional parameters

```
.../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"

.../class/ATOMCLIENT.var:
# atom00 is the Beowulf master server
hserver=atom00
bserver=atom00
```

Hard disk configuration

- In `/fai/disk_config/` a config file called 4GB exists:

```
# <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
```

Features of the hard disk configuration

- Multiple hard disks can be specified in one config file

- Easy specification of size, mount point and options

- Fixed or variable partition size

- User data on a partition can be preserved

- Option for mkfs and mount, e.g. nosuid, ro, reiserfs

- Windows partitions can also be included to `/etc/fstab`

- Automatic generation of `/etc/fstab`

Software configuration

Installing software is done by the a small Perl script which uses the Debian command `apt-get`.

Example for the class `COMPILE`:

```
# COMPILE: packages for developing software
PACKAGES install
task-c++-dev task-debug
bin86 m4 g77 byacc cvs
```

- Actions: install, remove, taskinst

- Package dependencies are solved

- `dpkg --get-selections` format is also possible

Example for configuration scripts

```
.../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
```

Cfengine example

```
#! /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"
 }
```

Install templates of config files

- Many configuration files have prepared templates

- Each installation needs a different template

- A class determines which template to use

- Copying of these files using `fcopy`

- Extracting of archives using `ftar`

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

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

Hooks

- The installation process is divided in several tasks (eg. define variables, partition local disk, install software, mount Debian mirror, call config scripts, save log files, . . .)

- For each default task, you can define one or more hooks

- Hooks are selected by class names

- Hooks are called before the default task action is performed

- Hooks can extend a task or replace the default task by skipping it

- It is very easy to customize FAI to your local needs with hooks

Information about FAI

- Homepage: `www.informatik.uni-koeln.de/fai`

- CVS repository, very active Mailling list, examples of log files

- Additional software available during installation: lvm, raidtools, dump, restore, ext2resize, hdparm, parted, resize_reiserfs

- Access to Debian mirror via NFS, FTP oder HTTP

- FAI with action `sysinfo` can be used as a rescue system. It mounts all local partitions automaticly using `/etc/fstab` if available

- Successful tests on SUN SPARC

Summary

- Fully unattended installation ! No interaction needed

- Identical, consistent installations are guaranteed

- Easy creation of configuration using classes

- Constant disk space on install server

- Central repository of all configuration and log files are centrally stored on the install server

- Quick and easy reinstallation produces a cluster with up to date software, less security holes

- **FAI is a scalable method for installing Debian GNU/Linux**