FAI – The Universal Deployment Tool

Thomas Lange, University of Cologne

lange@informatik.uni-koeln.de

LCA Systems Administration Miniconf, January 2015
finger lange@localhost

- whoami
  - Diploma in computer science, University of Bonn, Germany
  - Sysadmin since over two decades
  - SunOS 4.1.1 on SPARC hardware
  - Solaris Jumpstart
  - Started FAI in 1999
  - 1999 first cluster (16 × Dual PII 400 MHz)
  - Debian developer since 2000
  - Several talks and tutorials:
    - Linux Kongress, Linuxtag, DebConf, SANE, LCA, FOSDEM, CeBit, OSDC, UKUUG, FrOSCon, Chemnitzer Linuxtag
What is a deployment?

- FAI = Fully Automatic Installation
- Making a computer ready to work
- From power-off to applications running
- It’s all about software packages
- Installation and configuration
- Central administration and control
What is FAI?

- FAI does everything a sysadmin (you!) has to do, before users can log in to a brand new computer for the first time
- Server based tool for a script based automatic installation
- Installs and configures the OS and all applications
- No master or golden image needed
- Class system provides modularity
- Flexible and easy to expand with hooks
- It can’t plan your installation :-( but
- **Plan your installation and FAI installs your plan! :-)**
The configuration is stored on the install server

The installation runs on the client
Plan your installation!
PXE boot (DHCP, TFTP)
Install client runs as diskless client (aufs for rw access)
Define Classes and variables
Parts of an installation II

- Create partitions on local hard disk
- Create file systems
- Install software packages (OS and applications)
- Configure and customize packages (using scripts)
- Boot new system
The class concept of FAI

- You can group a list of hosts by using a class
- A host may belong to multiple classes
- Examples: FAIBASE GRUB DESKTOP GNOME demohost LAST
- Order of the classes defines the priority from low to high
- All parts of the installation are using the classes
The config space

|-- class/
|  |-- 10-base-classes
|  |-- 50-host-classes
|  |-- FAIBASE.var
|  `-- GERMAN.var

|-- disk_config/
|  |-- FAIBASE
|  |-- DESKTOP
|  `-- foobar04

|-- basefiles/

|-- package_config/
|  |-- FAIBASE
|  |-- DESKTOP
|  |-- GERMAN
|  |-- GNOME
|  `-- server07
Defining classes

Example: ...

```bash
#!/bin/sh
dpkg --print-architecture | tr a-z A-Z # AMD64, I386

case $HOSTNAME in
demohost)
    echo "FAibase DHCPC DEMO" ;;
gnomehost)
    echo "FAibase DHCPC DEMO XORG GNOME";;
esac

case $IPADDR in
    123.45.6.*) echo "CS_KOELN DESKTOP NET_6" ;;
esac

ifclass -o AMD64 I386 && echo "GRUB"
lspci | grep -q MATROX || echo "MATROX"
```
Variables

Example: .../class/FAIBASE.var:

FAIALLOW_UNSIGNED=1
KEYMAP=de-latin1-nodeadkeys
UTC=yes
TIMEZONE=Europe/Berlin
ROOTPW='$1$kBn.MWcO.B$djxB38B7dMkplhJHPf2d1'
LOGUSER=fai
YPDOMAIN=dept-a

- Define your own variables
- Use the variables in .../scripts/*
Disk partitioning

Example: .../disk_config/FAIBASE:

disk_config disk1    preserve_always:8 fstabkey:uuid

primary   /         4G-10G     ext4  rw,noatime,errors=remount-ro
logical  swap        1G       swap  rw
logical  /var        1G-2G    ext4  rw  createopts="-L var -m 5"
logical  /tmp        1G-2%    ext4  rw  tuneopts="-c 0 -i 0"
logical  /home       5G-       ext4  defaults

- File systems: ext[2,3,4], vfat, xfs, ReiserFS, NTFS, brtfs
disk_config disk1
primary - 50-100 --
primary swap 1G swap sw
primary - 2G-10G --
logical - 0--
logical - 0--

disk_config disk2 sameas:disk1

disk_config raid
raid1 /boot disk1.1,disk2.1 ext4 rw
raid1 / disk1.3,disk2.3 ext4 rw,acl,user_xattr
raid1 disk1.5,disk2.5 --
raid1 disk1.6,disk2.6 --

disk_config lvm
vg volg1 md2,md3
volg1-usr /usr 8G ext4 rw createopts="-O dir_index,resize_inode"
volg1-var /var 2G ext4 rw createopts="-O dir_index,resize_inode"
volg1-hl /home/local 10G ext4 rw,acl,user_xattr,noexec,nosuid,nodev
volg1-es /export/sites 3G ext4 rw createopts="-O none"
volg1-v /vservers 8G ext4 rw createopts="-O ~dir_index,~resize_inode"
Software package installation

Example: .../package_config/BEOWULF:

```bash
# packages for Beowulf clients

PACKAGES aptitude
    fping ganglia-monitor

    lam-runtime lam4 lam4-dev libpvm3 pvm-dev mpich
    scalapack-mpich-dev

PACKAGES install BEOWULF_MASTER
    gmetad apache
```

- Supported package tools: aptitude, apt-get, smart, rpm, urpmi, y2pmsh, yast, yum, zypper
Scripts and files

|-- scripts/
 |  |-- FAIBASE/
 |  |  |-- 10-misc Bourne shell script
 |  |  |-- 30-interface Bourne shell script
 |  |  `-- 40-misc Cfengine script
 |  `-- DEMO/
 |     |-- 10-misc Perl script
 |     `-- 30-demo Cfengine script
|
|-- files/
 |  `-- etc/
 |     `-- X11/
 |        `-- xorg.xonf/
 |           |-- FAIBASE
 |           |-- MATROX
 |           |-- CAD
 |           `-- demohost

` fcopy /etc/X11/xorg.conf
#! /bin/bash

# create NIS/NONIS config

fcopy -M /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
        ainsl -av /etc/yp.conf "ypserver $s"
        # don’t do this! # echo "ypserver $s" >> $target/etc/yp.conf
    done
fi

ainsl -v /etc/fstab "${hserver}:/home /home nfs ro 0 0"
ainsl -av /etc/default/ssh 'SSHD_OPTS=-4'

fcopy -Mv /etc/hosts.allow /etc/hosts.deny
fcopy -M /etc/X11/xorg.conf
## Installation Times

<table>
<thead>
<tr>
<th>Host, RAM</th>
<th>Software</th>
<th>Zeit</th>
</tr>
</thead>
<tbody>
<tr>
<td>E5-2690v2, 3.0 GHz, 128GB</td>
<td>5.4 GB</td>
<td>7 min</td>
</tr>
<tr>
<td>Core i7, 3.2 GHz, 6GB</td>
<td>4.3 GB</td>
<td>7 min</td>
</tr>
<tr>
<td>Core i7, 3.2 GHz, 6GB</td>
<td>471 MB</td>
<td>77 s</td>
</tr>
<tr>
<td>Core2duo, 2 GHz, 2GB</td>
<td>4.3 GB</td>
<td>17 min</td>
</tr>
<tr>
<td>Core2duo, 2 GHz, 2GB</td>
<td>471 MB</td>
<td>165 s</td>
</tr>
<tr>
<td>Pentium 4, 3 GHz, 1GB</td>
<td>2200 MB</td>
<td>10 min</td>
</tr>
<tr>
<td>Pentium 4, 3 GHz, 1GB</td>
<td>1100 MB</td>
<td>6 min</td>
</tr>
<tr>
<td>Pentium 4, 3 GHz, 1GB</td>
<td>300 MB</td>
<td>105 s</td>
</tr>
</tbody>
</table>

- New Cluster: 36 node, each Gbit, server with 10Gbit
- No change of the installation time (426 sec)
- Max. CPU usage on the server: system < 13%, user < 1.5%
- 10 Gbit network was saturated for 1 minute (98%)
- NFS is **NOT** a bottleneck
The universal tool

debian  ubuntu

Scientific Linux
Installing different distributions

- Booting FAI and disk partitioning does not need modification
- You can use a Debian nfsroot when installing CentOS
- Use a different base file for each distribution (rinse)
- Different access to package repository (sources.list, yum.repos.d)
- Adjust package names
- Adjust customization scripts

|-- basefiles/
  |-- CENTOS6_32.tar.xz
  |-- CENTOS6_64.tar.xz
  |-- CENTOS7_64.tar.xz
  |-- SLC6_64.tar.xz
  |-- UBUNTU_1410.tar.xz
The universal tool

- FAI does not distinguish between
  - bare metal
  - virtual host
  - chroot
  - LiveCD
  - Golden image

- It's always about installing and configuring software packages
- chroot: fai dirinstall
- chroot does not have a hard disk
- chroot does not need a kernel
- TODO: fai-cloudimage
- Maybe: fai-stack ;-) 
- FAI runs on i386, amd64, IA64, SPARC, PowerPC, ALPHA, z10 mainframe
FAI users

- Anonymous, financial industry, 32,000 hosts
- LVM insurance, 10,000 hosts
- City of Munich, 16,000 hosts
- Albert Einstein Institute, 1,725 hosts
- Zivit, 260 hosts on two IBM z10 EC mainframes
- Archive.org, 200+ hosts
- XING AG, 300-400 hosts
- Opera Software, ~300 hosts
- Stanford University, 450 hosts
- MIT Computer science research lab, 200 hosts
- The Welcome Trust Sanger Institute, 540 hosts
- Deutsches Elektronen-Synchrotron, 273 hosts
- Mobile.de, ~600 hosts
- Electricité de France (EDF), 1,500 hosts
- BUF, digital visual effects company, 1,000 hosts
- ETH Zurich, systems group, ~300 hosts
- StayFriends, 700+ hosts
- Grml, creating eight different ISOs, daily builds
<table>
<thead>
<tr>
<th>hostname</th>
<th>confdir</th>
<th>defclass</th>
<th>partition</th>
<th>extrbase</th>
<th>debconf</th>
<th>instsoft</th>
<th>configure</th>
<th>tests</th>
<th>savelog</th>
<th>faiend</th>
<th>reboot</th>
</tr>
</thead>
<tbody>
<tr>
<td>demohost</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>atom03</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>atom02</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>atom01</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>gnomehost</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
FAL - Fully Automatic Installation

FAL is a non interactive system to install, customize and manage Linux systems and software configurations on computers as well as virtual machines and chroot environments, from small networks to large-scale infrastructures like clusters and cloud environments.

It's a tool for unattended mass deployment of Linux. You can take one or more virgin PC's, turn on the power, and after a few minutes, the systems are installed, and completely configured to your exact needs, without any interaction necessary.

Motto: Plan your installation, and FAL installs your plan.

NEWS

- [28 Nov 2014] New FAL CD image available, FAL 4.3.1 - wheezy1
- [19 Nov 2014] FAL 4.3.1 released, bug fixes
- [24 Oct 2014] FAL 4.3 released, btrfs support added
- [3 Jun 2014] FAL 4.2 released, new ISO images created
- [15 September 2014] CentOS and Scientific Linux CentOS support [more...]
- [21 Dec 2009] The FAL project celebrates its 10th anniversary

Features

- Installs and updates Debian, Ubuntu, CentOS, RedHat, SUSE, ...
- Centralized deployment and configuration management
- Installs virtual machines using KVM, XEN or VirtualBox and Vserver
- Easy setup of software RAID and LVM
- Full remote control via ssh during installation
- Integrated disaster recovery system
- Every stage can be customized via hooks

Questions?