

Automatic Installation and System Updates with FAI

Overview, Functionality, Possibilities

Henning Sprang

Silpion IT Solutions GmbH/ Linux Solution Park GmbH

2008-07-3

The speaker - Henning Sprang



- FAI team member
- Software developer, author, admin, consultant
- Interests: Virtualization, systems management, Java, Web, QA
- Free Software enthusiast since first contact - about 1996

Overview

- 1 About systems installation
- 2 About FAI
- 3 Functions
- 4 Usage of FAI
- 5 Outlook

Some Questions...

What do you do...

- When your main server did explode?
- When a developer needs a new test system?
 - With EXACTLY the same config as the production system!
- When you bought yourself a new laptop?
- When you have to change Apache settings on a web cluster?

About system installation - Needs



- Need to recover system after a crash
- Need to reproduce systems exactly for testing
- Need to install multiple systems, fast
- And after install: you need to manage changes!

About system installation - how?



- Manual installation has some downsides:
 - Unreproducible - Humans make errors
 - Unefficient - install many systems takes a long time
- Solution: Automation
- Side effect: System setup is documented in code/config

disk images

.img?

- Create templates from “proper” installations
- Copy when needed and adjust them as needed
- Pros:
 - Low learning cost
 - Simple and fast implementation – cp/rsync/tar, README
- Cons:
 - Inflexible - the smallest change requires rebuilding the image
 - Still manual work needed to get a installed system
 - Storage cost linear to number of different configurations

Installation with shell scripts

`#!/bin/bash`

- Manual work replaced by shell scripts
- Pros:
 - Much lower storage cost than images
 - Higher flexibility
 - Tailor made
- Cons:
 - Full-blown development project (you'll realize after a while)
 - You solve every problem on your own, instead of reusing work of others

Usage of available auto-installers

Solutions are available. . .



The most important projects and target distributions:

- Anaconda und Kickstart: Fedora-based
- Autoyast: SuSE-based
- Nlite/Unattend: Windows in many flavours
- FAI: Debian-based, Fedora-based, SuSE-based, Mandriva, Windows, Solaris
- Others: Solaris Jumpstart, RedHat Cobbler/Koan

Why FAI?



- Flexible and easy to extend
- Simple, powerful architecture – “everything is a shell script”
- Open development model
- Community support seasoned „Installers“
- Diverse client- and server- distributions
- Multiple installation types and system update
- Can be used for real hardware and virtualization systems

History

- Started 1999 by Thomas Lange at the University of cologne
- Base idea: structured and planned installation
- „Plan your installation, and FAI installs your plan“
- Part of the Debian Distribution
- Today about 10 active developers, small but nice community
- Since 2005 softupdates from Henning Glawe included

Who is using it for what?



- EDF uses FAI (with GOSA) for some research clusters
- LiMux in Munich: installs/updates 400(to be 14000) clients/servers
- Multiple top 500 High Performance Clusters
- Small home networks starting from 3 systems
- GRML admin live CD built with FAI

Overview I



- FAI classes
 - A class defines system properties and actions to be taken
 - Class-assignment with simple texfile, database, or scripts
 - Systems can be assigned to multiple classes, combined at will

Overview II



- Server-distribution: Debian-based (Dependencies Perl, NFS, TFTP, debootstrap - easy to port)
- Target-distributions: Redhat, Debian, Ubuntu, SuSE, Mandriva, Fedora, Windows
- Different installation types (networked, CD/USB, chroot)
- Integrated versioning with subversion, CVS, git
- Softupdate for updates

Installation types



- Network installation with central install server
 - Client/Server architecture
- Directly calling dirinstall for chroots
- dirinstall for chroots and virtualization - xen-tools, ganeti)
- fai-cd / fai-usb (environments without network)
- With grml-live: Live-CD generation!



The most important FAI *tasks*/steps of installation:

- *defclass*: Class definition of the target system
- *partition*: Partitioning
- *extrbase*: Unpack a minimal base image
- *debconf*: read and apply Debconf preseedings
- *instsoft*: Software package installation
- *configure*: Run configuration scripts
- *savelog*: Push logfiles onto the install Server

Considerations – installation



- Decide the matching install type (net/cd/dirinstall)
- Plan your installation
 - Use cases
 - Network and environment
 - Software-packages
 - Additional files and config adjustments
- Using mirrors of Internet software repositories

Considerations - Updates



- How and when should which patches be applied?
- Testing processes - where do I know from, which effect a patch/update has?
- Mirrors of security update repositories?
- Automatic(regular, timebased) or manual updates?

FAI setup and configuration I



- Installation on Debian:
 - *apt-get install fai-quickstart*
- Adjust install server setup in /etc/fai:
 - fai.conf: LOGUSER=fai, LOGPROTO=ssh for Logging via ssh
 - apt/sources.list: use local mirror if available

FAI setup and configuration II



- FAI server is configured - create NFSroot:
 - *fai-setup*
- For PXE-Boot: *fai-chboot* to set boot-kernel and options
- Without PXE: *make-fai-bootfloppy*
- Installation from CD: *fai-cd*

Setup infrastructure services



For network install:

- DNS entry for server and clients
- DHCP config - Host/IP/MAC as usual
- FAI-specific:

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

Create System configurations



Configurations for installation are stored in FAI *configspace*

- Simple text files and scripts
- The requirements from the installation plan are reflected here
- Example included in `/usr/share/doc/fai/examples/simple`
- Default location: `/srv/fai/config`

The FAI configspace



Contents of configspace:

basefiles
class
debconf
disk_config
files
hooks
package_config
scripts

Adjust configspace - class

- *class* contains class- and variable definitions.
- Simplest way: assign classes based on hostnames
- Some sample classes: FAISERVER, GNOME, DEMO, XORG
- Any script can be used to assign classes
- E.G.: check specific hardware, MAC or IP, disk size, ...

Adjust configspace – basefiles

- Contains minimal base images for special uses or non-Debian distributions
- At the start of the install, this directory is checked for images for defined classes
- You could also put an image here and skip the rest :)

Adjust configspace – disk_config

Define harddisk configuration and mount points as needed

```
#<type> <mountpoint> <size mb> [mount options][;extra options]
```

```
disk_config disk1
```

```
primary    /          150-300    rw,errors=remount-ro ; -c -j ext3
logical     swap       40-500    rw
logical     /var       90-1000   rw                ; -m 5  -j ext3
logical     /tmp       50-1000   rw                ; -m 0  -j ext3
logical     /usr       200-4000   rw                ; -j ext3
logical     /home      50-        rw,nosuid         ; -m 1  -j ext3
# logical    /home      preserve9  rw,nosuid         ; -m 1  -j ext3
```

Adjust configspace - debconf



- Presets for package install scripts
- Only for dpkg-based distributions
- Works analog to Debian Installer

Adjust configspace - package_config

- Contents: files named by class names
- Purpose: Define packages to be installed
- Supports many installation methods:
 - install (apt-get)
 - aptitude
 - taskinst (Debian tasks=Package collections)
 - urpmi (mandriva)
 - yum (Fedora)
 - y2i (SuSE y2pms)
 - yast (SuSE yast -i)
 - Some more special ones...

Adjust configspace - package_config II

Example package_config/DEMO from 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
```

Adjust Configspace - scripts

- scripts contains scripts to be executed after package installation
- Usually shell-, Perl- and cfengine-scripts
- Need for others: just install Interpreter in the NFS-Root
- Naming scheme:
 <CLASSNAME>/<NUMBER>-<SCRIPTNAME>
- Number defines order of execution
- SCRIPTNAME arbitrary just for readability

scripts example:

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

Adjust configspace - files



- Structure of a filesystem, starting with /
- For usage with fcopy/ftar = classbases copy/unpack
- Copy single files explicitly, or recursive from / in a DEFAULT script

Adjust configspace - files II

- Target file=directory
- actually copied source file=CLASSNAME

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

Adjust configspace - hooks

- Hook naming scheme: `.<CLASSNAME>[.source]`
- Execution before the according task
- Optional `skip_task`: The actual task will not be executed, for example for skipping partitioning of Xen Domains
- Example:

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

Do the Installation

Depending on chosen install type:

- Boot via PXE/Bootfloppy/install-CD/USB Stick
 - PXE-config can be set with `fai-chboot`
- Start Xen VM with `xm create vm-name.cfg install=1`
- `fai dirinstall <TARGETDIR>` into mounted blockdevice
- Call `dirinstall` in xen-tools, Ganeti, ...
- Start system with `fai-cd/fai-usb`
- Run `grml-live`

Run System

- Depending on number of packages system is installed in 3-30 minutes (Server/Desktop)
- Restart with production configuration
- Functionality tests

Outlook

- Support stuff for more distributions: *fai-distributions*
- GOSA as LDAP and FAI GUI
- Automatic Tests of the installed systems:
 - hooks/scripts could check files and configurations
 - Crucible Test Framework
- No LDAP? Management-Tool/lightweight GUI without GOSA

Further informations

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

Questions?

Questions?