Automated Linux Management Infrastructure

ease of installation, central configuration, scalable and sustainable administration

Arcady Genkin
agenkin@cs.toronto.edu

http://www.cdf.toronto.edu/~agenkin/downloads/

Department of Computer Science

University of Toronto

May 11, 2005

Presentation Outline

- Introduction into context
- Design principles (objectives)
- Alternatives for Linux installation
- Concept of Classes
- FAI overview and examples
- Cfengine overview and examples
- Putting it all together
- Security considerations
- Gripes
- Future plans
- Questions welcome at any time

Context: CDF Labs

CDF (Computing Disciplines Facility).
Serving Computer Science students at the Department of Computer Science.

- Over 3000 student accounts
- 10 rooms in two buildings
- Over 200 Linux workstations
- 6 Linux compute servers
- 4 Linux infrastructure servers
- 40 X terminals
- 2 Solaris infrastructure servers
- 5 FTE staff

Context: Hardware

- Dell GX260
- Dell Precision 340
- Dell GX150
- Dell GX110
- Eurocom L287S
- No-name clone workstations
- Dell and OSS servers

Context: Operating Systems

- Red Hat Linux 7.3
- Windows 2000

Need an acceptable platform replacement for RH 7.3 (obsoleted) and Win2K (painful to maintain).

- RHEL (Enterprise Linux) ordeal

Context: Former Linux installer setup

- RedHat's Kickstart
- One Kickstart configuration file per hardware/purpose combination
- Installation boots off floppy:
  - need to initiate installation in person
  - need to update BIOS before and after installation
- After installation all configuration manual in-place
- Rely on back-ups in case of failure

Note: not meant as criticism of the Kickstart installer; our Kickstart configuration was probably suboptimal.

Objectives

Want a Linux management system:

- Easy to configure and administrate
- Few ties to a particular distribution
- Flexible for local customisations
- Easier to reinstall than fix a broken installation
- Decouple machine’s function from machine’s hardware
- One shared installation configuration for all hosts

Requirements: Installation System

- Unattended installation
- Remotely triggered re-installation (PXE boot)
- Customisable (custom hooks, scripts, etc.)
- Easy to configure for a large set of hardware combinations
- Easy to maintain (new software/hardware)
- Can invoke the Configuration System (next slide) from within

Requirements: Configuration system

- Centralised
- Secure
- Easy to use
- Platform-independent

Requirements: Linux distribution

- University-frendly licensing
- Automated installer
- Automated networked updates
- Local mirror

Alternatives: Linux Distributions

- RedHat Enterprise Linux
- SuSE Linux
- Debian
- More?..

Alternatives: Installers

- Image-based (disk duplication) installers
  - Norton Ghost / g4u (Ghost for Unix)
  - Simple to understand
  - Can work in broadcast mode
  - Need one image per hardware/software configuration
  - Need a lot of space to keep history of changes
- Smart installers
  - FAI / Kickstart
  - Harder to set up
  - Heavier burden on the network during simultaneous installation
  - The most flexibility

Alternatives: Configuration systems

- Home-grown
- Cfengine
- More?..

Classes: The concept

- A machine is either in a class or not
- Classes are attributes
- A machine can be in many different classes
- Hardware / software (usability) / run-time classes
- Keep software classes task-oriented
- Make your own script to determine most classes
- Some “built-in” classes the software determines automatically

Classes: Examples

- DELL_GX260 NET_E1000 VIDEO_RADEON SOUND_INTEL_8X0
- VMWARE NET_AMD_PCNET32 VIDEO_VMWARE
- CDFBASIC / CDF_COMPUTE_SERVER
- RESTRICTED_SHELL_ACCESS
- DNSCACHE_SERVER
- CDF_NTP_SERVER
- GRUB

FAI: Fully Automated Installation

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

- Mostly Debian specific
- Booting flexibility (PXE/floppy/CD-ROM)
- Classes determined by a script at install time
- Works by means of NFS-rooted environment
- Hooks at any point during installation
- Virtually any software available during installation
- Optional VT/SSH for debugging
- Uploads installation logs over SCP/FTP
- High level configuration

FAI: Disk partitioning example

```
# File disk_config/CDF_WORKSTATION

disk_config hda

#primary         -   preserve1

primary /boot  50   rw          ; -j ext3
logical /      100  rw          ; -c -j ext3
logical swap   1024 rw
logical /usr   2000 rw          ; -m 0 -j ext3
logical /var   1000 rw          ; -m 5 -j ext3
logical /tmp   1000 rw,nosuid ; -m 0 -j ext3
logical /local 4500 rw          ; -m 0 -j ext3
logical /data  0-   rw,nosuid ; -m 0 -j ext3
```

FAI: Packages configuration example

```
# File package_config/CDF_WORKSTATION

PACKAGES install

# Network-related stuff
ssh dnsutils whois finger nfs-common
nfs-kernel-server
qmail-under-slash-local-cdf

# Development stuff
gcc g++ gcc-2.95 g++-2.95 gcc-3.0 g++-3.0 g77
g77-2.95 g77-3.0 gcj gcj-3.0 gettext ddd
gdb make patch rcs cvs python mit-scheme

# and so forth ...
```

Cfengine: Configuration engine

`http://www.cfengine.org/`

- Centralised architecture
- High level configuration files syntax
- Supports classes
- Open Source / Multi-platform
- Secure (private/public key cryptographic authentication)
- Robust (authenticate securely, transmit unencrypted)
- Can copy files, create symlinks/directories, check/fix permissions, run shell commands, edit files, mount directories, etc.
- Can monitor system, email notifications, etc.
- Ensures sustained state of configuration

Cfengine examples: warning

Important: The following Cfengine snippets have been modified for the demonstration purposes and to fit into the slides.

Cfengine: `cf.nfs`

```
control:
  actionsequence = ( directories files links )
directories:
  any::
    /h
    /s

files:
  any::
    /h  mode=755 owner=root group=root
    /s  mode=755 owner=root group=root

links:
  any::
    /u -> /cdf/u
```

Cfengine: `cf.fstab`

```
control:
  actionsequence = ( copy )

copy:
  cdf_workstation.!unique_fstab::
    /export/debian/etc/fstab.cdf_workstation
                destination=/etc/fstab
                server=$(policyhost)
                mode=444 owner=root group=root

  unique_fstab::
    /export/debian/etc/fstab.$(host)
                destination=/etc/fstab
                server=$(policyhost)
                mode=444 owner=root group=root
```

Installation: The big picture

Bits and pieces:

- Manually installed FAI host
- DHCP, TFTP, NFS servers
- Central Cfengine configuration server (policy server)
- Full local Debian mirror, exported over NFS or HTTP (use `apt-mirror`)

Installation: Sequence

- configure PXE for reinstall and reboot host
- PXE Boot
- Mount root over NFS
- FAI bootstraps (determines classes, etc.)
- FAI partitions hard disk
- FAI installs minimal base system
- FAI chroots and installs additional packages + updates

(Continued on next slide...)

Installation: Sequence (continued)

- script bootstraps Cfengine
- script chroots and runs Cfengine
- Cfengine fine-tunes the host just like normal
- script checks for success of installation
- script updates PXE server for normal boot next time
- FAI saves all logs using SCP

Takes 10 minutes to install a host, including syncing 3GB of `/local`.

Security considerations

- Disable VT and SSH during FAI installation
- Pros and cons of PXE booting
- Initial exchange of Cfengine’s public keys

Gripes: FAI

- Documentation is complete, but sometimes awkward
- Simultaneous installation of more than 10 hosts strains the network and NFS servers (not FAI's fault)
- Newer versions of FAI can only install *Debian/Sarge*, older version can only install *Debian/Woody*
- Can't easily have several NFS roots with older versions of FAI

Gripes: Cfengine

- Initial set up rather difficult
- Documentation is extensive, but lagging behind software development
- File server inefficient for synchronising large (several gigabytes) directory trees: had to resort to `rsync` over SSH
- Interfaces with RPM but not Debian package manager (yet?)

Other useful Debian stuff

- `apt-mirror`
  - `http://apt-mirror.sourceforge.net/`
  - Platform-independent
  - Only needs Perl and `wget`
  - Can mirror Debian security repository
  - Syntax very close to `sources.list`
- Debian backports project
  - `http://www.backports.org/`
  - Backports of packages from Debian/testing into Debian/stable

Plans for future improvements

- Start using Cfengine's monitoring and notification capabilities
- Test FAI installer with *Debian/Sarge* or *Ubuntu*
- Maintain separate NFS-rooted installation environments
- Look at available options for securing PXE booting