New Generation of Linux Meta-Installers

FOSDEM Research Track (25.2.2007)

Paulo Trezentos (Paulo.Trezentos@caixamagica.pt)
Roberto DiCosmo
Stéphane Lauriere
Mario Morgado
João Abesacis
Fabio Mancinelli
Arlindo Oliveira
Agenda

About Caixa Mágica...

About EDOS...

Meta-installers
  - Dependency solving
  - Rollback
  - Hardware support (briefly)
About Caixa Mágica Software

Linux distribution since 2000

Company created in 2004

*Open source specialist*

- 15 to 20 employers
- Located in Lisbon center

Large projects

- Warner cinema network – 200 POS
- Ministry of Justice (3 departments)
- Ministry of education (14,000 dual boot computers)
About EDOS

Environment for the development and Distribution of Open Source software.

IST project

EDOS is a research project funded by the European Commission as a STREP project under the IST activities of the 6th Framework Programme. The project involves universities – Paris 7, Tel Aviv, Zurich and Geneva Universities –, research institutes – INRIA – and private companies – Caixa Magica, Nexedi, Nuxeo, Edge-IT and CSP Torino.
What is a installer? And a meta-installer?

Installer:
- rpm / dpkg
- Problems: not solve dependencies, command-line

Meta-installer:
- Apt-get, Apt-rpm, URPMI, YUM, Yast, Smart
- Is your meta-installer capable of solving all the needed dependencies?
- Can you rollback a installation? And a removal?
- Can you use P2P to install a package? Why not?
The New Generation of meta-installers

What would you like to have in your meta-installer?

- Roll-back
- (Almost) Perfect dependency solving
- P2P package retrieving (Radu Pop presentation)
- Hardware support distribution

Method: research and apply to Apt-get / Apt-rpm the results
Meta-installer and EDOS

WP2 address the problem of dependency solving:
- Is the rollback / history of the same difficulty?
- Caixa Mágica developed Apt-rpm rollback feature
- See next slides for results
- **Final stage**

WP2 proposed a SAT solving mechanism for server side dependency checking
- Is it also possible to apply it for client side?
- See next slides for the analysis
- **Research stage**

Hardware support distribution capabilities
- Is it possible to distribute hardware support through Meta-installer
- **Already happens**
I – Dependency problem

Meta-instalers – Problem:

- Dependency solving is a NP-Complete problem (proofed by WP2 team – Paris 7)
- Use some simple (apt, portage,...) or not so simple (smart) heuristics
- Client-side is a bit out of the scope of EDOS project
The “car” problem
Apt approach

```
root@sclara # apt-get install car

The following packages have unmet dependencies:
car: Depends: wheel (>= 2) but it is not going to be installed
E: Broken packages
```

Fail to install directly
If you force it it install wells
Portage approach

```
z1on cm-test # emerge -pv --tree cm-test/car

These are the packages that I would merge, in reverse order:

Calculating dependencies ... done!
[blocks B ] =cm-test/tyre-2 (is blocking cm-test/glass-2)
[ebuild N ] cm-test/car-1 0 kB [1]
[ebuild N ] cm-test/door-2 0 kB [1]
[ebuild N ] cm-test/window-2 0 kB [1]
[ebuild N ] cm-test/glass-2 0 kB [1]
[ebuild N ] cm-test/wheel-3 0 kB [1]
[ebuild N ] cm-test/tyre-2 0 kB [1]
[ebuild N ] cm-test/engine-2 0 kB [1]

Total size of downloads: 0 kB
```

And more...

- If we force to install windows-1 it stills fail since it tries to update it before install car
SAT Solver approach
Conversion into DEP–SPEC

- DEP car ALL engine wheel door
- DEP engine ONE engine1 engine2
- DEP engine1 ALL turbo
- DEP turbo ONE turbo1
- DEP wheel ONE wheel2 wheel3
- DEP wheel3 ALL tyre
- DEP tyre ONE tire1 tire2
- DEP door ONE door1 door2
- DEP door2 ALL window
- DEP window ONE window0 window1 window2
- DEP window1 ALL glass
- DEP window2 ALL glass
- DEP glass ONE glass1
- DEP glass ONE glass
BLIF

.model t1.blif
.inputs engine1 engine2 turbo1 wheel2 wheel3 tire1 tire2 door1 door2 window0 win dow1 window2 glass1 glass2
.outputs install engine wheel door window
.names engine wheel door car
111 1
.names engine1_vout engine2 engine
10 1
01 1
.names turbo engine1_vin
1 1
.names engine1 engine1_vin engine1_vout
CNF

c Generated by blif2cnf 1.0 (jpms@ecs.soton.ac.uk).
c Copyright (c) 2005–2006 Joao Marques-Silva.
c

c Variable Mapping:
c turbo-1-0 -> 57
c window-2-0 -> 22
c wheel-2-0 -> 3
c glass-1-0 -> 14
c wheel-3-0 -> 39
c glass-2-0 -> 15
c tyre-1-0 -> 34
c car-2-0 -> 2
SAT solver – Minisat

blic  unit reducing...2 unary clauses identified.
c  resolving...34 resolutions performed.
c  unit reducing...0 unary clauses identified.
c  resolving...0 resolutions performed.
c Processing phase stats:
c  Initial clause count       : 133
c  New clause count           : 116
c  Seconds elapsed (real time): 0 seconds.
c Learn order: 3
c Fudge factor: 0.9
c Solution phase timeout after: 43200 seconds.
c Finding all solutions...
Solution 1: 1 2 4 6 12 14 15 17 22 24 26 27 28 30 32 33 34 37 38 39 40 41 42 43 44 45 46 47 48 52
Solution 2: 1 2 4 12 14 15 17 22 24 26 27 28 30 32 33 34 37 38 39 40 41 42 43 44 45 46 47 48 52
Solution 3: 1 2 4 6 12 14 15 17 22 24 26 27 28 29 31 33 34 37 38 39 40 41 42 43 44 45 46 47 48 52
Solution (using minimax)

**Solution 1**
- c ROOT install -> 1
- c car-2-0 -> 2
- c glass-1-0 -> 14
- c glass-2-0 -> 15
- c window-2-0 -> 22
- c window-0-0 -> 24
- c door-2-0 -> 30
- c tyre-1-0 -> 34
- c wheel-3-0 -> 39
- c engine-1-0 -> 43
- c engine-2-0 -> 44
- c ROOT window -> 48

**Solution 3**
- c ROOT install -> 1
- c car-2-0 -> 2
- c glass-1-0 -> 14
- c glass-2-0 -> 15
- c window-2-0 -> 22
- c window-0-0 -> 24
- c door-1-0 -> 29
- c tyre-1-0 -> 34
- c wheel-3-0 -> 39
- c engine-1-0 -> 43
- c engine-2-0 -> 44
- c ROOT window -> 48
Future work – dependency solving

Define what is a good solution:
- The most recent packages?
- Less packages?
- Less change in RPM installed packages?

Define the SAT/PBO approach to it

Integrate in Apt–RPM a basic mechanism

Research:
- Multi-criteria constraint solving
- Pseudo-Boolean Optimization (PBO)

Test it and compare
II – Rollback / history

Proposed to do in scope of EDOS WP2 (Paris meeting – November 2005). Work of David Pinheiro, João Abecasis, Paulo Trezentos, ...

Completely different from “rpm –repackage”

rollback–hist

- Display package installation/removal history

rollback

- Rollback to a specific state

Configuration is stored in /var/cache/apt/pkgcfg

- apache2–cfg66–2.0.50–7.2.tar.bz2
Apt–RPM rollback
E.g.: `apt-get rollback 11`
*(rolling back the removal of a package)*

1 – Query the database

APT-RPM

2 – Install the needed RPM

RPM Repository

3 – Apply the previous configuration

Configuration (tarball)
`/var/cache/apt/pkgcfg`
Rollback contributions & Dissemination

Apt–rpm
- Patch was submitted (29–11–2006)
- 2nd patch was submitted (30–01–2007) with “configure” capability to disable “rollback”
- Maintainer confirmed intention to integrate it in next releases
- SVN: http://aptrpm.caixamagica.pt/repo/aptrpm/branches/rollback@aptrpm

Apt–get
- Patch was submitted (26–12–2006)
- SVN: http://aptrpm.caixamagica.pt/repo/aptrpm/branches/rollback
Future work – rollback

Fine grain in the rollback

Allow a snapshot even without a package operation

Rollback all filesystem changes

- pre/post install scripts change outside the rpm file set
- Possible solution: using a different file system (UnionFS,...)
III – Hardware support...

SuSE Kernel Module Packages (KMP)
- Kernel symbols encoded as dependencies in KMP packages
- Kernel symbols encoded as provides in kernel packages

DELL's DKMS
- Complementing RPM package system
- Support to RHAS, SLES,...

Two different problems:
- Distribute new kernel modules
- Identify a non-support hardware and trigger the associated RPM package
Conclusions
Meta-installer can be enhanced in 4 areas:

- Rollback
- Dependency solving
- Kernel module (and userland apps) support
- P2P package installation

APT can be a good testbed for some of EDOS research achievements

Direct Impact on apt-\{get/rpm\} and indirect on the rest
Thank you.

Paulo.Trezentos@caixamagica.pt