

Massive scale installation and management tools for Debian GNU / Linux (part 3)

Author: André Felipe Machado<andremachado@techforce.com.br>

Having 50 thousand machines across the country to install and manage for your enterprise is not a trivial task.

This is the third part of this text.

Many references are listed on the parts 1 and 2 of this text.

Large scale installations and management

A strong, clever and healthy development and user community around an open source project is the long term key differentiator for the success that we took into account for this analysis.

Some clever projects stalled or are in low pace mode without these.

Without a large enough user base, not enough bugs could be ironed out.

Also, the "bus factor" must be evaluated. If a project is pushed by very few developers, an unexpected departure could cause a big impact or a complete stall.

There is an interesting discussion thread about the configuration management [here](#)

There is an excellent series of articles about automatic large scale installations, configuration, deployment, management of Debian systems [here](#)

If you are short on time, and must choose only one book about large enterprise systems administration, read "The Practice of System and Network Administration", and the author of [\(67\) Sure, you can find here](#) systems and networks sysadmins and managers (also CIOs and even CEOs) could benefit too.

You should, also, lurk for some time at least at [The League of Professional Systems Administrators](#) opportunities reading about real life experiences.

Preliminary Conclusions (October 2007)

It is dated because most of the key tools choices should be evaluated again in 6 months or less. The key projects are very active and fast paced, as most of them reached critical mass. In the open source, software libre FLOSS environment, 6 months is enough time to a whole landscape change.

You will likely use a combination of tools. FAI leveraging debian-installer and debconf preseeding, then Puppet for the hard configuration and management tasks, and GOsa2 for the daily tasks.

Ebox could be used for some of the machines and as it will soon release a multiserver management version, should be evaluated again.

But remember that GOsa is able to manage servers and workstations by now and it is field proven.

Puppet is able to provide management, so after you master its features and language, you could even leave graphical tools like GOsa2 behind or for daily regular staff tasks. Bcfg2 is focused at configuration, not management by now.

FAI

Fully Automatic Installation, [FAI](#)

THE massive scale installation tool.

Before the debian-installer and debconf preseeding [0] [25], it was the unique also.

~~FAI does not have (16 September 2007) | Fitati |~~ the debian-installer dealing with some schemes of partitioning.

Now, in the best "software libre" tradition, it works **WITH** debian-installer and debconf.

Evolving since 1999, it is a mature and solid software, field proven at massive installations, with a healthy development and user community around it.

It can work even with other distributions, but shines with Debian, where it born.

It is a very powerful tool, and you will have a learning curve ahead. But for enterprise wide installations, it is essential and light years ahead of other installation and deployment tools.

As it is not a simple image, cloning tool, it can install at similar but not exactly equal systems, at the same profile. Multiple profiles can be configured.

When used with the debian-installer and debconf, your work will become simpler and at an abstraction layer above.

Could be used as a configuration management tool, but there are other ones more suitable for this huge task.

An European automatic installations report

A presentation about enterprise installations with Debian [39].

An interesting article [55].

Puppet

[Puppet](#) [5] is something like a descendant of ISconf and was created by experienced sysadmins to substitute the honorable cfengine.

Leveraging the years of experience using cfengine, its idiosyncrasies were addressed and a brand new elegant architecture conceived.

It is a declarative description configuration tool, modular, extensible, with a vibrant and healthy development and user community around it, accelerating its development.

More and more users are participating, but it is still not so mature as cfengine.

Also, an external (to the Puppet project) worry: as of this writing, the Ruby interpreter is still showing [obscure effects](#) affecting Puppet at some scenarios [40].

A new Ruby [41] shows (bugs, but) still lags behind the venerable rock solid perl interpreter. I doubt Ruby versions 1.8.x and below could solve these bugs at mainline releases. Perl was born in 1987 [born in 1995](#) niche language until 2004 [with the Rails framework](#)

Because cfengine has its idiosyncrasies, and lack of development community (it is a single dictator open source development model, with no external contributors source code direct commits, no public repository), has a very clear declarative description programming, modular, extensible, plugins, and a complete new architecture was created leveraging years of cfengine use, it is worth the hassle of carefully setting up an EXCLUSIVE (virtual) server(s) with bare minimal software needed for it and the most simple configuration attainable.

Actually, Puppet was created clearly with cfengine technical and community shortcomings in mind and even a [direct comparisor](#)~~[45]~~ exists

There are modules for different package managers.

Puppet community, development speed and acceleration are very strong. As of August 2007, there are users declaring more than 1000 machines managed.

A graphical interface for SOME of its features and tasks is in the works (as of August 2007) and could make GOsa or Ebox unneeded at some time in future (after enough debugging).

The puppet engine is very powerful and flexible. Many of its features and tasks could be more complicated and convoluted to be used through a graphical interface than from the command line and configuration / declaration files.

As these advanced features should be in charge of experienced sysadmins, not afraid of command line and configuration files, they are likely to stay without a graphical interface. At least until a human interface expert comes with a clever solution for such tasks.

An intr~~[47]~~oduction to Puppet article, [part 1](#) [part 2](#)

A [report](#) using Puppet [48].

An [intervi](#)~~[50]~~ew of the lead developer [49]. Another [interview](#)

A video [presentation](#) at Linuxconf Australia 2007 about Puppet [51].

Bcfg2

[Bcfg2](#) [6] is another declarative description configuration management tool created to address the cfengine shortcomings.

It uses xml files for the declarations.

Does not have a so healthy and vibrant development community around it and much of its development is controlled by a single developer, but not so tightly as cfengine.

Written in perl it does not suffer interpreter obscure bugs.

There is [link \[52\]](#) and [link \[53\]](#) extensive comparison between bcfg2 and puppet starting at [this link \[54\]](#) to a [link \[55\]](#) discussion at [this other](#)

You should evaluate again each 3 months, comparing to Puppet, as it is a fast paced project too.

Cfengine

[Cfengine](#) [3] is a venerable, mature, solid software configuration management tool, field proven at massive deployments.

It is a procedural configuration tool.

But it has many idiosyncrasies, architecture concept limitations, and lack of development real community. There is, actually, a single developer who accepts or rejects suggestions and code contributions sent to him.

Written in perl, it is not safe interpreter obscure bugs.

Puppet and bcfg2 were created with cfengine shortcomings in mind, so they still not have cfengine extensive track record.

There is an extensive comparison of Puppet X Cfengine at [this link](#) [45]

If you are in a hurry, simply could not afford any bug (application or interpreter), and MUST use rock solid field proven software now, despite the shortcomings ([puppet X cfengine comparison](#) [45]) then you should use FAI with Cfengine.

There are a very interesting discussion thread [here](#) [54].

GOsa2

[GOsa2](#) [10] is a web based system to manage accounts and systems in LDAP databases.

It can be used with FAI and provide automated installation of systems using profiles, for example.

Written in php, it has been used field proven, mostly in Europe government agencies, with their large installations.

The good integration with FAI and LDAP is a significant feature.

Munich is using GOsa to manage 14 thousand desktops and servers.

GOsa is good for daily large networks and user management tasks, but, today, a graphical interface is not suitable for massive and scriptable tasks.

So, using Puppet for these tasks is more suitable, and using GOsa (or Ebox) for the remaining and for less experienced sysadmins could be a good approach. Until Puppet graphical interface for some tasks reaches stable release.

In the state of things at 2007, no graphical interface concept available is precise enough, smart enough, scriptable enough to accomplish all complex sysadmin tasks. But some are good tries.

Read the [complete article](#) and be careful with the heading and batantly wrong.

There [is a good article regarding GOsa here](#)

Ebox

[Ebox](#) [11] is a web based system to manage accounts, network services and systems and can be used with LDAP databases.

Its scope is broader and different than GOsa, reaching more network services, servers, software, and even backup, content filtering and traffic shaping. It started as single machine management tool. The new version under development (October 2007) is able to central management of various ebox servers. Things will become interesting.

But does not manage installations, yet, making it a good complement for Puppet, bcfg2, FAI, but not integrated with them.

You could use it after you installed and configured your machines.

Written in perl language, does benefit from a solid interpreter. Very modular.

The development is accelerating, a healthy community is building up around project and soon may reach critical mass, relying on even more developers.

There is a simple tutorial about installing Ebox on Debian [here](#)

Other tools:

Hyperic

[Hyperic](#) [60], the [multipleservering](#) like [Zabbix](#) tool, and with configuration features. But is still somewhat bogus and with limitations yet, but under very heavy development and must be tracked and evaluated again each 3 months. Having a company behind it, contract supports are offered. Hyperic has an active user base at the open source community site

OpenQrm

[OpenQrm](#) [12] is more like a system imager management tool, focused on rpm distro based servers and its deployment and management.

NetDirector

[NetDirector](#) [15] is a rpm distro based massive configuration and deployment tool, written in Java, and using LDAP.

ISconf

[ISconf](#) [17] is something of a grandparent of Puppet, and it is in slow mode now.

AIS

[AIS](#) [8] is an interesting project and concept, but stalled and needs a new maintainer group. Read the [presentation](#) [55] about AIS.

Oceano

[Oceano](#) [16] is a pre-alpha project, but should be tracked.

Psgconf

[Psgconf](#) [18] is an interesting configuration tool, but stalled.

LCFG

[LCFG](#) [7] is an interesting configuration tool and active project, focused on rpm based distros and Solaris. Some efforts for port it to Debian is underway, but not officially yet.

Landscape

[Landscape](#) [14] is a commercial remote web service offered by Canonical, for monitoring and management of servers, users.

SmartFrog

[SmartFrog](#) [61] is a handy idea focused on java grid computing configuration management.

StateEngine

[StateEngine](#) [62] is another interesting concept, but it is at slow pace and at early stages for production environments.

More tools

There is [another article listing configuration management tools at wikipedia at this link](#) [63]

References:

The remaining references are listed on the parts 1 and 2 of this text.

[38] <http://pascal.case.unibz.it/handle/2038/1038>

[39] <http://www.ringworld.org/~dieman/talks/asia-miniconf05/>

[40] <http://reductivelabs.com/trac/puppet/wiki/FrequentlyAskedQuestions#what-s-with-these-damned-segfaults>

[41] <http://antoniocangiano.com/2007/02/19/ruby-implementations-shootout-ruby-vs-yarv-vs-jruby-vs-gardens-point-ruby-net-vs-rubinius-vs-cardinal>

[42] <http://history.perl.org/PerlTimeline.html>

[43] http://en.wikipedia.org/wiki/Ruby_language

[44] http://en.wikipedia.org/wiki/Ruby_on_Rails

[45] <http://reductivelabs.com/trac/puppet/wiki/CfengineVsPuppet>

[46] <http://www.debian-administration.org/articles/526>

[47] <http://www.debian-administration.org/articles/528>

[48] <http://blog.bureado.com.ve/?p=400>

[49] <http://www.open-management.com/2007/5/21/interview-with-luke-kanies-of-puppet>

[50] <http://computerworld.com.au/index.php?id=224515074>

[51] <http://www.linux.org.au/conf/2007/talk/84.html>

[52] <http://www.madstop.com/puppet-vs-bcfg2.html>

[53] <http://changelog.complete.org/posts/519-Thoughts-on-cfengine,-bcfg2,-and-puppet.html>

[54] <https://lists.ubuntu.com/archives/ubuntu-devel/2005-April/006862.html>

[55] http://www.usenix.org/events/lisa04/tech/full_papers/mikhailov/mikhailov_html/index.html

[56] <http://hyperic-hq.sourceforge.net/>

[57] <http://useopensource.blogspot.com/2007/08/linux-administration-will-become-gui.html>

[58] <http://www.onlamp.com/pub/a/onlamp/2004/12/02/gosa.html>

[59] <http://www.debianhelp.co.uk/ebox.htm>

[60] <http://www.zabbix.org>

[61] <http://www.smartfrog.org>

[62] <http://linuxczar.net/moin/StateEngine>

[63]
http://en.wikipedia.org/wiki/Comparison_of_open_source_configuration_management_software

[64] <http://www.linuxhelpforum.com/showthread.php?t=812826>

[65] <http://people.debian.org/~az/>

[66] [http://en.wikipedia.org/wiki/C_\(programming_language\)](http://en.wikipedia.org/wiki/C_(programming_language))

[67] <http://everythingsysadmin.com/>

[68] <http://lopsa.org>

[69] <http://www.isissetup.ch>

[70] <http://taktuk.gforge.inria.fr>

[71] <http://taktuk.gforge.inria.fr/kanif/>

[72] <http://git.or.cz>

[73] <http://kitenet.net/~joey/code/etckeeper/>

[74] http://kitenet.net/~joey/blog/entry/announcing_etckeeper/