SPM without SPMA

Simplifying software package management with Yum

Luis Fernando Muñoz Mejías

HPC group, DICT, Universiteit Gent

14^{th} Quattor Workshop
29-31/10/2012

Outline

What we always thought we wanted

- Strict package management
- Only what is listed is installed
  - And we list it with painful detail
- Select the repository from which each package must come from
- All operations are atomic

... and we actually got it before anyone else

What we always thought we wanted

- Strict package management
- Only what is listed is installed
  - And we list it with painful detail
- Select the repository from which each package must come from
- All operations are atomic

... and we actually got it before anyone else

What we paid for that

- Dependency hell
- Complicated update workflows
  - swrep-soap!!
  - check-deps!!
- Code bloat
  - In the client code
  - In the node descriptions
- Package bloat
  - So we are not that strict after all
- Overengineered design
  - We thought we'd be able to extend it, but we can't

Figure: How it actually feels

What we've learned to want

- Automatic dependency resolution is paramount
- We don't need to specify the exact versions and architectures for **each single package**
- But we want to be able to do so for selected ones
- But we can't afford any intrusive changes

Alleviating our problems with Yum

- Dependency resolution
- Ability to lock some versions
- Ability to specify loosely some other packages
- Proxy support
- Out-of-the-box package signature support
- Tools to decide what has to be cleaned up

... how good does it sound?

Alleviating our problems with Yum

- Dependency resolution
- Ability to lock some versions
- Ability to specify loosely some other packages
- Proxy support
- Out-of-the-box package signature support
- Tools to decide what has to be cleaned up

... how good does it sound?

Upgrade concerns

- No concerns for the user, really
  - Just have the correct versions of Yum (3.2.29) and yum-utils (1.1.30-10)
- Upgrade as with any other component
- Well, we still need AII support

This works even better with a demo

Let's chop off some members (of our CDBs)

So I updated, what's next?

```
template  components/spma/repository_cleanup;

"/software/packages" =
  resolve_pkg_rep(value("/software/repositories"));
"/software/repositories" =
  purge_rep_list(value("/software/packages"));
```

- Remove the body of the template
  - Removes the largest hot spot in our compilation
  - Profile size in an ordinary CE is reduced by ~10%

So I updated, what's next?

```
template  components/spma/repository_cleanup;

"/software/packages" =
  resolve_pkg_rep(value("/software/repositories"));
"/software/repositories" =
  purge_rep_list(value("/software/packages"));
```

- Remove the body of the template
  - Removes the largest hot spot in our compilation
  - Profile size in an ordinary CE is reduced by ~10%

So I updated, what's next?

```
template  components/spma/repository_cleanup;

"/software/packages" =
  resolve_pkg_rep(value("/software/repositories"));
"/software/repositories" =
  purge_rep_list(value("/software/packages"));
```

- Remove the body of the template
  - Removes the largest hot spot in our compilation
  - Profile size in an ordinary CE is reduced by ~10%

Installing packages is easier now

- The old pkg_* functions still work
- But they are too verbose now
- And slow
- I like this one better

```
"/software/packages/{gcc}" = nlist();
```

Death to swrep-soap-template and update.repository.templates!!

- Repository templates are **constant** now

```
structure template repository/dag_el5;
"name" = "dag_el5";
"owner" = "admin@domain";
"protocols" = list(
  nlist("name","http",
        "url","http://something") );
```

Less Pan code

- No more package bloat
- Just declare the packages you need
- I bet we'll install only 1/3 of the packages we currently have
- I bet we'll **declare** in the profiles less than 1/3 of that

Obsoleted packages

- `SPMA`
  - Replaced by Yum
- `rpmt-py`
  - Replaced by Yum

Ordinary upgrades

- Edit templates only for packages that have their versions locked
- Just run `yum distro-sync` in a cron job
- Or enable the nightly upgrades
- No more errata templates
- But needs smarter repository management in some institutions

In short...

- The new Yum backend kills **a lot** of Perl and Pan code
  - Faster and simpler operations
- We need better repository management
  - Managing one repository is easier than managing 1200 packages
- Three missing things
  - All support
  - GPG key support
  - Package blacklisting
- Expected in production at UGent by end of November

Questions?