Ultimate Debian Database: data mining Debian made easy!

Lucas Nussbaum

debian

FOSDEM 2009

Debian : the data hell

A lot of different sources of data in Debian

With different data formats :
text files, BerkeleyDB, SQL databases, ...

Need to combine them all :
Mainly for Quality Assurance, e.g :

- *Packages of priority $\geq$ standard with RC bugs ?*
- *Maintainers with lots of outdated/buggy packages ?*

Ultimate Debian Database

Idea :

- **Import all the data in a single (Postgre)SQL DB**
- Easier to query (relatively well-known interface)
- *The* proper way of joining data together
- No need to write problem-specific scripts
  - e.g the *how many packages in Debian ?* thread on Planet

History

Started as a Google Summer of Code project in 2008

Student : Christian von Essen (Neronus)

Mentors :

- Lucas Nussbaum (lucas)
- Marc Brockschmidt (HE)
- Stefano Zacchiroli (zack)

Results :

- Very good work from Christian
- Usable code at the end of the summer
    - mostly Python, some Perl

Design choices

Not problem-specific, no typical queries
(not `projectb` or the new wanna-build DB !)

Schema :

- Typical user == human
- Make it easy to write/run queries
- Performance ? important, but not a critical goal
- No *surrogate keys*

Surrogate key

- Unique identifier (usually integer)
- Used as primary key
- Not derived from any application data

packages (**package_id**, package_name, ...)

MySQL : AUTO_INCREMENT
PostgreSQL : serial

Has both advantages and disadvantages

Details :
`http://en.wikipedia.org/wiki/Surrogate_key`

Design choices (2)

Data :

- Correctness is critical
- Partial updates ? Often difficult/risky
- Solution : complete data reloads
  - Using transactions to avoid temporary unavailability

Design choices (3)

Debian is inconsistent

- What does "package" mean ?

Inconsistency can be interesting for QA

⇒ Keep inconsistency in UDD
⇒ No foreign keys between data sources

Current status

- Hosted on **`udd.debian.org`** (aka `piuparts.cs.helsinki.fi`)
  - German cabal hosted `udd.debian`**`.net`**
- Uses PostgreSQL 8.3 (required)
- You can connect from `{merkel,alioth,master}.d.o`
  - e.g : `/usr/lib/postgresql/8.3/bin/psql service=udd`
  - Even non-DDs can connect !
  - Please don't D.O.S UDD !
- Schema / data is semi-stable

More info :

`http://wiki.debian.org/UDD`

What we currently import

All those sources are imported using :

- Working and monitored scripts
- Ran regularly (cron jobs or ssh triggers)

Sources and Packages data

Imported from the Sources and Packages files

- From Debian and Ubuntu

Almost all fields are imported

- Except typos, and non-standard fields : Orginal-Maintainer, OrigianI-Maintainer, Originalmaintainer, Build-Recommends, Npp-*, Gstreamer-*, Vdr-patchlevel, ...

Sources and Packages data (2)

```
select distribution, release, count(*) from all_sources group by
distribution, release order by distribution, release;
```

distribution	release	count
debian	etch	10558
debian	etch-proposed-updates	40
debian	etch-security	257
debian	experimental	1062
debian	lenny	12562
debian	lenny-proposed-updates	65
debian	lenny-security	17
debian	sid	13477
debian-backports	etch	496
debian-volatile	etch	8
ubuntu	hardy	14302
ubuntu	intrepid	15131
ubuntu	jaunty	15505

Sources and Packages data (3)

Packages table : one row per (package, version, architecture, distribution, release, component)

⇒ 534724 rows ! (only Debian, Ubuntu is a different table)

⇒ `packages_summary` table without architecture information

Debian bugs

- BTS : file-based storage
  `/org/bugs.debian.org/spool/db-h/56/129956.summary`
- Use the Debbugs perl module
- Performance problem :
  need to open and read many small files
  Use `posix_fadvise(2)` ?
  - Allows to predeclare an access pattern

  Preload all files into cache ?

Currently :

- Importing unarchived bugs : 27 minutes
- Importing archived bugs : 82 minutes

Debian bugs (2)

We also compute :

- the corresponding source package
- `affects_(stable|testing|unstable|..)`

```
udd=> select count(*) from bugs where
affects_testing and severity >= ’serious’;

 count
-------
   128
(1 row)
```

Ubuntu bugs

Another proof that proprietary software is evil.

Two usable "APIs" :

- Text version of bugs

  `https://bugs.launchpad.net/bugs/231402/+text`
- REST API (recent)

Problem with both :
Fetch several bugs with a single HTTP request ? no !

Ubuntu bugs (2)

"Solution" :

- Run several workers in parallel that fetch bug data
- Worked *too* well : DOSed Launchpad with 8 workers
- Slowed down. Takes about 2 hours every day

Bug to ask for a text export of all bugs in a single file : LP #231402

Popularity Contest

- For both Debian and Ubuntu
- Also `popcon_src` : popularity contest for source packages

```
popcon(package, insts, vote, olde, recent,
nofiles)
popcon_src(source, insts, vote, olde,
recent, nofiles)
```

Lintian

Imported from `http://lintian.debian.org/lintian.log`

```
lintian(package, tag_type, package_type, tag)
```

Most "popular" Lintian tags :

```
select tag, tag_type, count(*) from lintian
where tag_type != 'information'
group by tag, tag_type
order by count desc limit 10;
```

Lintian (2)

tag	tag_type	count
debhelper-but-no-misc-depends	warning	5289
symlink-should-be-relative	warning	3694
binary-without-manpage	warning	3628
image-file-in-usr-lib	overriden	3329
manpage-has-errors-from-man	warning	2857
executable-not-elf-or-script	warning	2756
copyright-without-copyright-notice	warning	2719
image-file-in-usr-lib	warning	2648
manpage-section-mismatch	warning	2309
out-of-date-standards-version	warning	1822

Debtags

Imported from the Debtags SVN repository

```
debtags(package, tag)
```

- Multiple rows for multiple tags

package	tag
bash	uitoolkit : :ncurses
bash	suite : :gnu
bash	scope : :utility
bash	role : :program
bash	interface : :shell
bash	implemented-in : :c

Carnivore

Carnivore

- Database about identities used by maintainers
- Links logins, emails, PGP keys
- Used by MIA

```
carnivore_emails(id, email)
carnivore_keys(id, key, key_type)
carnivore_login(id, login)
carnivore_names(id, name)
```

Upload history

- Work done by Filippo Giunchedi (godog)
- Generated from `debian-devel-changes@` archives

```
upload_history(id, package, version, date,
changed_by, maintainer, nmu, signed_by,
key_id, fingerprint)
```

```
upload_history_architecture(id, architecture)
```

```
upload_history_closes(id, bug)
```

Uploads that closed the most bugs

package	version	count
glibc	2.2-7	159
ifupdown	0.6.5	144
apt	0.3.14	132
aptitude	0.4.0-1	131
gcc-3.3	1 :3.3ds9-1	127
glibc	2.3.5-3	125
xfree86	4.3.0-1	116
gcc-4.0	4.0.1-1	113
apt	0.5.5	101
gcc-4.1	4.1.1-8	98
manpages	1.58-1	90
apt	0.5.0	79

Orphaned Packages

- Convenience table
- Built from wnpp bugs
- Determine when a package was orphaned by parsing the bug log

```
orphaned_packages(source, type, bug,
description, orphaned_time)
```

Testing migrations

- Built by tracking testing's Sources files
- *When did that package last migrate to testing ?*
- Data since 2005, using `snapshots.d.n`

What’s missing ?

- Status wrt upstream (DEHS) (hi Raphael !)
- Build status (Wanna-build)
- Britney output
- Inactive maintainers (MIA)
    - Possible privacy issues — currently only DD can access it
- Minor changes to the schema (?)

So, what can we find out about Debian using UDD ?

Source-only uploads, anyone ?

- Upload without any architecture-specific package
  - only source and arch : all packages
- Already possible !
- Who does it ?

Uploads :

- Of arch : any source packages
- With only arch : all packages

Source only uploads, anyone ? (2)

```
SELECT package, version, signed_by
FROM upload_history uh
WHERE package in
 (SELECT source FROM sources
 WHERE distribution = ’debian’ AND release = ’sid’
 AND architecture = ’any’)
AND NOT EXISTS
 (select * from upload_history_architecture uha
 WHERE uh.id = uha.id
 AND uha.architecture NOT IN (’all’,’source’))
ORDER BY date DESC;
```

Source only uploads, anyone ? (3)

package	signed_by
git-core	Gerrit Pape
zsh-beta	Clint Adams
xtables-addons	Pierre Chifflier
kfreebsd-7	Aurelien Jarno
pyenchant	Piotr Ozarowski
fbasics	Dirk Eddelbuettel
mseide-msegui	Torsten Werner
llvm	Pierre Habouzit
ircd-hybrid	Aurélien GEROME

...

Number of different lintian errors

```
select package, count(distinct tag) from lintian
 where tag_type = ’error’ group by package;
```

Number of different lintian errors

```
select package, count(distinct tag) from lintian
 where tag_type = 'error' group by package;
```

package	count
gcc-snapshot	7
apache2-mpm-itk	4
harden-doc	4
nws	4
sgml-base-doc	4
openswan	4
gallery	3
euro-support	3
gallery2	3

gcc-snapshot lintian errors

```
select tag, count(*) from lintian
 where tag_type = 'error' and
 package = 'gcc-snapshot' group by tag;
```

gcc-snapshot lintian errors

```
select tag, count(*) from lintian
 where tag_type = ’error’ and
 package = ’gcc-snapshot’ group by tag;
```

tag	count
unstripped-binary-or-object	1
wrong-path-for-interpreter	1
malformed-override	1
invalid-arch-string-in-source-relation	1
build-depends-indep-without-arch-indep	1
python-script-but-no-python-dep	1
symlink-contains-spurious-segments	7

Let’s look at lenny !

(and at more positive things)

Who uploaded lenny’s packages ?

```
select changed_by, count(*) from sources s, upload_history uh
where s.source = uh.package and s.version = uh.version
and s.distribution=’debian’ and s.release = ’lenny’
group by changed_by order by count desc limit 5;
```

Who uploaded lenny’s packages ?

```
select changed_by, count(*) from sources s, upload_history uh
where s.source = uh.package and s.version = uh.version
and s.distribution=’debian’ and s.release = ’lenny’
group by changed_by order by count desc limit 5;
```

changed_by	count
gregor herrmann <gregor+debian@comodo.priv.at>	179
Christian Perrier <bubulle@debian.org>	144
gregor herrmann <gregoa@debian.org>	142
Daniel Baumann <daniel@debian.org>	136
Julien Cristau <jcristau@debian.org>	131
Matthias Klose <doko@debian.org>	123
Damyan Ivanov <dmn@debian.org>	116
Peter Eisentraut <petere@debian.org>	116
Dirk Eddelbuettel <edd@debian.org>	112

Who uploaded lenny’s packages ?

Using carnivore :

```
select distinct cn.name, count(*)
from sources s, upload_history uh,
carnivore_emails ce, carnivore_names cn
where s.source = uh.package and s.version = uh.version
and s.distribution=’debian’ and s.release = ’lenny’
and substring(uh.changed_by from ’<(.*)>’) = ce.email
and ce.id = cn.id
group by cn.name order by count desc limit 20;
```

Who uploaded lenny's packages ?

name	count
Gregor Herrmann	321
Christian Perrier	144
Daniel Baumann	136
Julien Cristau	132
Matthias Klose	126
Damyan Ivanov	116
Peter Eisentraut	116
Dirk Eddelbuettel	112
Varun Hiremath	104
Bart Martens	102
Barry deFreese	96
Marc Brockschmidt	92

Who uploaded lenny’s NMUs ?

```
select distinct cn.name, count(*)
from sources s, upload_history uh,
carnivore_emails ce, carnivore_names cn
where s.source = uh.package and s.version = uh.version
and s.distribution=’debian’ and s.release = ’lenny’
and substring(uh.changed_by from ’<(.*)>’) = ce.email
and ce.id = cn.id
and uh.nmu
group by cn.name order by count desc limit 20;
```

Who uploaded lenny's NMUs ?

name	count
Christian Perrier	127
Peter Eisentraut	65
Marc Brockschmidt	63
Luk Claes	60
Mark Hymers	52
Chris Lamb	46
Matthias Klose	43
Pierre Habouzit	37
Moritz Muehlenhoff	35
Petter Reinholdtsen	31
Barry deFreese	28
Amaya Rodrigo Sastre	25
Riku Voipio	25
Thomas Viehmann	24

Who closed lenny’s RC bugs ?

RC bugs closed amongst bugs reported since the release of etch (08/04/07)

```
select done, count(*) from all_bugs
where status = ’done’ and arrival >= ’2007-04-08’
and severity >= ’serious’
and submitter != done
and source in (select source from sources
    where distribution=’debian’
    and release = ’lenny’)
and substring(done from ’<(.*)>’) not in
 (select maintainer_email from sources
  where sources.source = all_bugs.source
  and distribution=’debian’ and release=’sid’
  union select email from uploaders
  where uploaders.source = all_bugs.source
 and distribution=’debian’ and release=’sid’)
group by done
order by count desc limit 30;
```

Who closed lenny's RC bugs ?

name	count
Steve Langasek <vorlon@debian.org>	81
Luk Claes <luk@debian.org>	60
Matthias Klose <doko@cs.tu-berlin.de>	59
Thomas Viehmann <tv@beamnet.de>	53
Pierre Habouzit <madcoder@debian.org>	51
Nico Golde <nion@debian.org>	49
Cyril Brulebois <cyril.brulebois@enst-bretagne.fr>	44
peter green <plugwash@p10link.net>	41
Riku Voipio <riku.voipio@iki.fi>	39
Ben Hutchings <ben@decadent.org.uk>	33
Julien Cristau <jcristau@debian.org>	32
Chris Lamb <chris@chris-lamb.co.uk>	32
Thijs Kinkhorst <thijs@debian.org>	31
Philipp Kern <pkern@debian.org>	27

Who reported lenny’s RC bugs ?

RC bugs reported since the release of etch (08/04/07)

```
select submitter, count(⋆) from all_bugs
where status = ’done’ and arrival >= ’2007-04-08’
and severity >= ’serious’
group by submitter
order by count desc limit 10;
```

Who reported lenny's RC bugs ?

name	count
Lucas Nussbaum	2045
Michael Ablassmeier	458
Bastian Blank	429
Matthias Klose	323
Kurt Roeckx	321
Frank Lichtenheld	217
Nico Golde	193
Daniel Schepler	150
Martin Zobel-Helas	127
Marc 'HE' Brockschmidt	109

<Insert your query here !>

Future work / Help needed !

We need help to :

- Play with UDD, build tools on top of the DB
  - So we know what's missing/should be improved
- Implement missing importers
  - mostly DEHS, wanna-build, britney, MIA
- Improve performance, if possible

Contact : #debian-qa or debian-qa@l.d.o

Thank you !

- Christian von Essen (GSOC student)
- Marc Brockschmidt, Stefano Zacchiroli (co-mentors during GSOC)
- German cabal - Dalug admins (hosting infrastructure for udd.debian.net)
- DSA, esp. Peter Palfrader, Martin Zobel (hosting for udd.debian.org)
- Everybody for help, feedback and constructive comments

Conclusion

UDD is ready, go play with it !

`http://wiki.debian.org/UDD`