

Guida allo sviluppo collaborativo

Riccardo Magliocchetti

whoami

Free software developer

- Maintainer: django-admin-bootstrapped, bootchart2
- contributore: uwsgi, LibreOffice

Oggi parliamo di Free Software

Libertà di:

- eseguire
- studiare e modificare
- ridistribuire l'originale
- ridistribuire le mie modifiche

Mani avanti

Free Software != Supporto gratuito

Nessuna garanzia :)

Come sono sviluppati i progetti?

Linux

- gerarchia: dev -> maintainers -> Linus
- review: mailing list
- bugs: mailing list, bugzilla
- sviluppo: master + feature freeze
- rilasci: ogni 2 mesi
- installazioni: miliardi
- sviluppatore tipo: pagato

LibreOffice

- gerarchia: orizzontale
- review: gerrit
- bugs: bugzilla
- sviluppo: master -> freeze -> branch
- rilasci: 6 mesi
- installazioni: milioni
- sviluppatore tipo: pagato

django-admin-bootstrap

- gerarchia: nessuna
- review: github
- bugs: github
- sviluppo: master
- rilasci: quando serve
- installazioni: migliaia
- sviluppatore tipo: volontario

Doocracy

Chi fa decide (di solito)

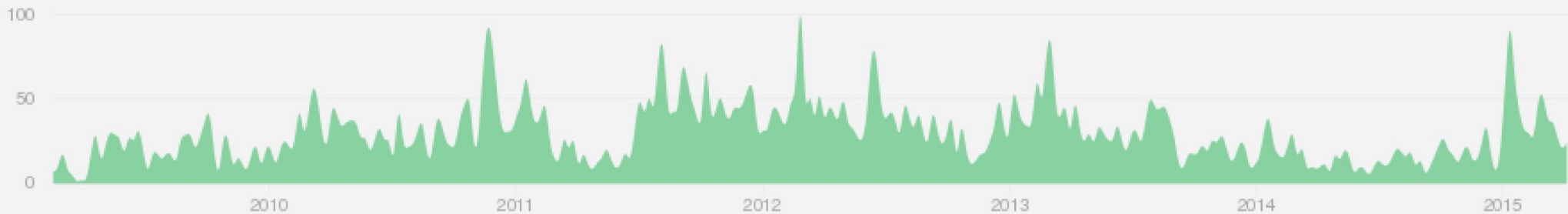
Come si dirimono le questioni? Buonsenso?

Questioni non tecniche

"Noi siamo a guardia della legge che vogliamo immutabile, scolpita nel tempo." (cit.)

- OpenOffice.org vs LibreOffice
- node.js vs io.js

node.js vs io.js in numeri



Come si collabora in pratica?

Strumenti: mailing list

Discussione su \$qualcosa riguardante il \$topic

Aiutati a farti aiutare:

- più contesto possibile
- netiquette

Strumenti: bug tracker

PREREQUISITI:

- ho idea di cosa sto facendo
- **NON** è lo strumento per il supporto

USARE SE:

- ho trovato un bug
- c'è qualcosa che potrebbe essere fatto meglio

Strumenti: irc

supporto, discussioni, socialità :)

Strumenti: stack overflow

- **NON** ho idea di cosa sto facendo
- cerco una consulenza gratuita

A chi appartiene il futuro?



Stephanie Vacher CC BY-NC-ND 2.0

Contribuire

Barriere di ingresso: Linux vs LibreOffice

Non solo codice: bug triaging, documentazione

FYI Roma non è stata fatta in un giorno

Bug reporting

- Cosa ho fatto
- Cosa è successo
- Cosa mi sarei aspettato
- Step per riprodurre

Feature requests senza patch



Stefano Petroni CC BY-NC-ND 2.0

Contribuire codice



Accettare i fallimenti



Scott Beale CC BY-NC-ND 2.0

Persistere

[GIT PULL] overlay filesystem v25

16 posts by 4 authors  



Miklos Szeredi

Other recipients: toiv...@linux-foundation.org, vi...@zeniv.linux.org.uk, linux-...@vger.kernel.org, linux-...@vger.kernel.org, linux-...@vger.kernel.org

Linus,

Please pull

[git://git.kernel.org/pub/scm/linux/kernel/git/mszeredi/vfs.git](https://git.kernel.org/pub/scm/linux/kernel/git/mszeredi/vfs.git) overlayfs.v25

Plenty of bugs fixed relative to the previous version, thanks to Al Viro's review and is now officially be BugFree(TM). Also passes the unionmount-testsuite by David Howells.

Thanks,
Miklos

Commit

- Un commit per cambiamento "logico"
- Non mischiare bugfix e cleanup
- performance? numeri!
- reference a discussioni e bug

Commit do: Linux

vfs: simplify and shrink stack frame of link_path_walk()

Commit 9226b5b440f2 ("vfs: avoid non-forwarding large load after small store in path lookup") made link_path_walk() always access the "hash_len" field as a single 64-bit entity, in order to avoid mixed size accesses to the members.

However, what I didn't notice was that that effectively means that the whole "struct qstr this" is now basically redundant. We already explicitly track the "const char *name", and if we just use "u64 hash_len" instead of "long len", there is nothing else left of the "struct qstr".

End result: fewer live variables in the loop, a smaller stack frame, and better code generation. And we don't need to pass in pointers variables to helper functions any more, because the return value contains all the relevant information. So this removes more lines than it adds, and the source code is clearer too.

Commit do: bootchart2

pybootchartgui: _parse_proc_ps_log rewrite with iterator

Iterators use much less memory, so larger bootcharts may be processed without triggering OOM killer and massive swapping.

On a (big) 11MB tarball this will have a performance penalty of about ~10% but consuming half the memory.

Before:

23.50user 1.20system 0:24.97elapsed 98%CPU (0avgtext+0avgdata 770048maxresident)k

After:

26.78user 0.44system 0:27.24elapsed 99%CPU (0avgtext+0avgdata 321192maxresident)k

Commit don't

first big (really bug) master refactoring

 master  2.0-rc1 ... 1.9

 authored on 12 Feb 2013

 Showing **10 changed files** with **729 additions** and **724 deletions**.

Commit don't

m

 **master**



4201321 authored on 15 Jul 2014



Showing **2 changed files** with **71 additions** and **27 deletions**.

Pull request

1 pull request \sim 1 feature

- Usate, scrivete, aggiornate i test
- Usate eventuali tool di linting (se richiesti dal progetto)

Pull request do: django

Commits on Mar 30, 2015



Update tests to stop leaking models in shared AdminSite ...

xrmx authored 11 days ago

Commits on Apr 7, 2015



Remove duplication between index and app_index methods ...

xrmx authored 28 days ago



Fixed #24553: Return the full app list in index, app index, changeform... ...

xrmx authored 11 days ago ✓

Pull request don't: bootchart2

🔗	🔗	FIX charts -- doubtful fix for divide by zero	a0a2a34
🔗	🔗	comment expanded	bcfdb81
🔗	🔗	TODO	3ca5e79
🔗	🔗	FIX do not overfill charts	74a77ef
🔗	🔗	mouse wheel drives xscale	64f128c
🔗	🔗	FIX -- show final thread name -- from prctl -- when taskstats is absent	d47244d
🔗	🔗	pruning -- add option to select policy -- default heavy	71fcd8c
🔗	🔗	FOLD -- pruning -- do NOT elide bootchart collector	ade8e3d
🔗	🔗	DOC group options	95cc0af
🔗	🔗	FOLD -- pruning -- do NOT hide really tiny processes	2872bd6
🔗	🔗	COMMENT only -- taskstats -- treat process labels as implicit prctl e... ...	d2eac5f
🔗	🔗	IO charts per partition -- generalize to sequence of charts 3.partition... ...	7aaa26b
🔗	🔗	IO charts per partition 9.partitions on command line	f627f8b
🔗	🔗	gui -- xscale cosmetics	f31d94c

Code review

Non siamo il nostro codice, accettiamo le critiche!

Git to the rescue

```
# sistemare ultimo commit  
git commit --amend
```

```
# lavorare sempre su un branch!  
# pick, edit, squash, shuffle  
git rebase --interactive $branch
```

```
# riscrivo la history  
git push -f
```

```
# butto via tutto quello fatto dopo $sha1  
git reset --hard $sha1
```

FLOSS al lavoro

Upstream first! (o almeno provateci)

"It's the duty of all Free Software developers to steal as much time as they can from their employers for software freedom."

Jeremy Allison

Happy Hacking :)

Riccardo Magliocchetti

riccardo@menodizero.it

@rmistaken

<http://menodizero.it>

<https://github.com/xrmx>