bup: Git for backups

#bup #28c3

Zoran Zarić

- @zoranzaric
- Computer Science student at TU Darmstadt
- bup since April 2010

toc

1. Motivation
2. Git backgrounds
3. bup
   - 3.1 Features
   - 3.2 Algorithms & data structures

Motivation

- Space efficiency of backups
- Convenient access to backups
- Safety against bitrot, filesystem-, and media errors
- Safety against history changes

Git

Git

- Distributed version control system

Git

- Distributed version control system
- Content addressed

Git

- Distributed version control system
- Content addressed
- Immutable objects

Git

- Distributed version control system
- Content addressed
- Immutable objects
- Snapshot- instead of diff-based

Git: A Repository

Git: A Repository

- BLOBs

e69de29

Hello World

Git: A Repository

- BLOBs

  e69de29

- Trees

  82e3a75

```
100644 blob 5e1c309dae7f45e0f39b1bf3ac3cd9db12e7d689 README
100644 blob 39c8418e04721b9a30232ce754cac8d9ee78340a DESIGN
040000 tree 482fa65ae85c1e5bca8c091b479de60b714a4b6a src
```

Git: A Repository

- BLOBs

  e69de29

- Trees

  82e3a75

- Commits

  3dfe461f

tree a3d703e579dc9baae20456eb63fa49f5e4e7c9b4
author Zoran Zaric <zz@zoranzaric.de>1314498536 +0200
committer Zoran Zaric <zz@zoranzaric.de>1314498536 +0200
Example commit

Git: A Repository

63866463d511a245a55a57ca48efe8e67b955dec

- BLOBs

  e69de29

- Trees

  82e3a75

- Commits

  3dfe461f

- Tags & Branches

  v0.1 master

Git: A Repository

- BLOBs

  e69de29

- Trees

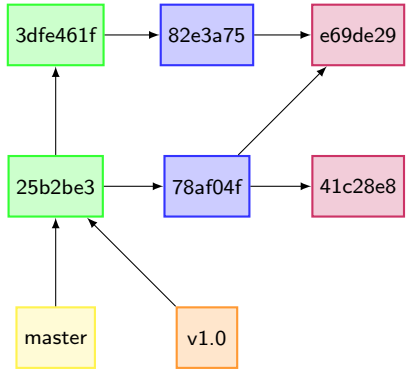
  82e3a75

- Commits

  3dfe461f

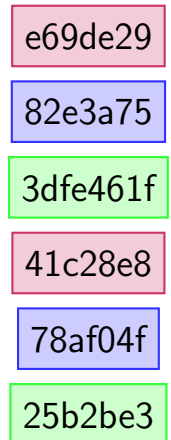
- Tags & Branches

  v0.1 master



Git: A Repository

- Packfiles



Git: Problems

Git: Problems

- Slow & memory-hungry for bigger files

Git: Problems

- Slow & memory-hungry for bigger files
- No meta data (permissions, owners, ACLs)

bup

- Avery Pennarun (git subtree, sshuttle, redo)
- `https://github.com/apenwarr/bup`
- `http://groups.google.com/group/bup-list`

bup: Installation

```
$ sudo apt-get install python2.6-dev python-fuse
$ sudo apt-get install python-pyxattr python-pylibacl
$ mkdir ~/src && cd ~/src
$ git clone https://github.com/apenwarr/bup.git
$ cd bup
$ make
$ make test
$ sudo make install
```

bup: Examples

```
$ bup index -ux /home/zz
$ bup save -n laptop /home/zz

$ bup save -r myserver -n laptop /home/zz

$ bup on myserver index -ux /home/zz
$ bup on myserver save -n server /home/zz

$ bup ls laptop/latest/home/zz
```

bup: Features

Deduplication (`http://goo.gl/aBpny`)

bup: Features

Deduplication (`http://goo.gl/aBpny`)

- Benchmark with two servers and a pseudo vm image on them with little changes
  rsnapshot: 4.97G
  bup: 2.18G

bup: Features

Deduplication (`http://goo.gl/aBpny`)

- Benchmark with two servers and a pseudo vm image on them with little changes
  rsnapshot: 4.97G
  bup: 2.18G
- Import of rsnapshot backups to bup
  rsnapshot: 12.6G
  bup: 4.6G

bup: Features

Meta data (almost done)

- Owner
- Exakt times
- Permissions
- Extended ACLs
- SELinux

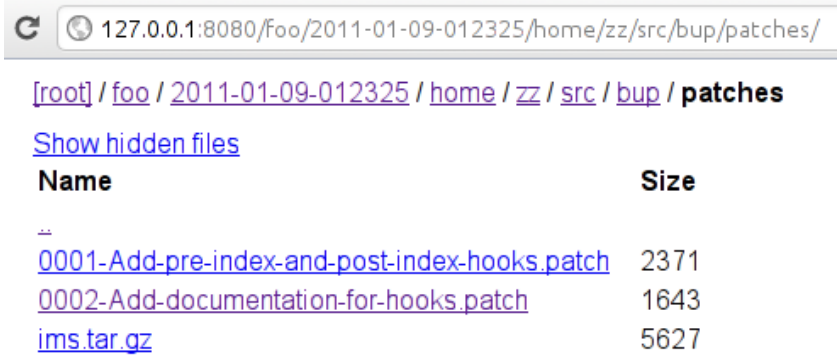
bup: Features

FUSE module

You can mount your backups and browse them with your favorite filemanager

bup: Features

Web interface



bup: Features

Runs on dd-wrt

bup: Features

Import-script for rsnapshot backups
More will follow (Duplicity)

bup: Features

Full compatibility with Git
Git tools like gitk or tig can be used with bup repositores

bup: Features

Uses par2 to be save against bitrot, filesystem-, and media-errors

bup: Algorithms & Data Structures

- Hashsplitting
- Midx
- Bloom filters

Hashsplitting

- Rolling checksum
- `rsync`'s algorithm
- Big files are split in 8kB Chunks (avg)
- 11 least significant bits of the checksum "1" $\Rightarrow$ new chunk

Midx

- idx: indexes for packfiles
- 1 idx per packfile
- An object is found with 3-4 lookups per packfile
- Midx for several packfiles
- Object is found with 2 lookups
- Problem: midx have to be recreated for every change

Bloom Filters

- Probabilistic data structure
- Check if a datum is known
- Append possible
- False-positives
  - Rate grows with added data
  - When rate >1% the bloom filter is expanded and rewritten
- Hash function optimized for few 1s in result
- Bloom filter is a bitarray; the result is added with bitwise OR
- When a hit is found a midx-lookup is done

Recent

- Meta data support about to be finished (patchset available, testing needed)
- Repack patches pending (deleting old backups)
- inotify based daemon is being discussed

You & bup?

- Python & a bit of C
- Native Windows support?
- OSX / Windows meta data support?
- OSX "inotify"-like port?
- GUI?
- Diff

Thank You

- @zoranzaric
- `zorzar` on freenode & hackint
- zz@zoranzaric.de (Email & Jabber)
- `zoranzaric.de`
- `github.com/zoranzaric`
- `gplus.zoranzaric.de`

- Slides: `zoranzaric.de/bup-28c3.pdf`