Software Heritage
Development & Operational Challenges to Preserve our Software Commons

Stefano Zacchiroli

Software Heritage
zack@upsilon.cc

10 March 2017
Incontro DevOps Italia
Bologna, Italy

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

(Free) Software is everywhere

Software source code is *special*

Harold Abelson, Structure and Interpretation of Computer Programs

"Programs must be written for people to read, and only incidentally for machines to execute."

Quake 2 source code (excerpt)

```
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalfs = 1.5F;

    x2 = number * 0.5F;
    y = number;
    i = * ( long * ) &y; // evil floating point bit level hacking
    i = 0x5f3759df - ( i >> 1 ); // what the fuck?
    y = * ( float * ) &i;
    y = y * ( threehalfs - ( x2 * y * y ) ); // 1st iteration
// y = y * ( threehalfs - ( x2 * y * y ) ); // 2nd iteration, this
can be removed

    return y;
}
```

Net. queue in Linux (excerpt)

```
/*
 * SFB uses two B[l][n] : L x N arrays of bins (L levels, N bins per level)
 * This implementation uses L = 8 and N = 16
 * This permits us to split one 32bit hash (provided per packet by rxhash or
 * external classifier) into 8 subhashes of 4 bits.
 */
#define SFB_BUCKET_SHIFT 4
#define SFB_NUMBUCKETS  (1 << SFB_BUCKET_SHIFT) /* N bins per Level */
#define SFB_BUCKET_MASK (SFB_NUMBUCKETS - 1)
#define SFB_LEVELS      (32 / SFB_BUCKET_SHIFT) /* L */

/* SFB algo uses a virtual queue, named "bin" */
struct sfb_bucket {
        u16             qlen; /* length of virtual queue */
        u16             p_mark; /* marking probability */
};
```

Len Shustek, Computer History Museum

"Source code provides a view into the mind of the designer."

Our Software Commons

Definition (Commons)

The commons is the cultural and natural resources accessible to all members of a society, including natural materials such as air, water, and a habitable earth. These resources are held in common, not owned privately. https://en.wikipedia.org/wiki/Commons

Definition (Software Commons)

The software commons consists of all computer software which is available at little or no cost and which can be altered and reused with few restrictions. Thus *all open source software and all free software are part of the [software] commons.* [...]

https://en.wikipedia.org/wiki/Software_Commons

Our Software Commons

Definition (Commons)

The commons is the cultural and natural resources accessible to all members of a society, including natural materials such as air, water, and a habitable earth. These resources are held in common, not owned privately. `https://en.wikipedia.org/wiki/Commons`

Definition (Software Commons)

The software commons consists of all computer software which is available at little or no cost and which can be altered and reused with few restrictions. Thus *all open source software and all free software are part of the [software] commons.* [...]

`https://en.wikipedia.org/wiki/Software_Commons`

Source code is *a precious part* of our commons

are we taking care of it?

Software is spread all around

Fashion victims

- many disparate development platforms
- a myriad places where distribution may happen
- projects tend to migrate from one place to another over time

Software is spread all around

Fashion victims

- many disparate development platforms
- a myriad places where distribution may happen
- projects tend to migrate from one place to another over time

Where is the place …

where we can find, track and search *all* source code?

Software is fragile

Like all digital information, FOSS is fragile

- inconsiderate and/or malicious code loss (e.g., Code Spaces)
- business-driven code loss (e.g., Gitorious, Google Code)
- for obsolete code: physical media decay (data rot)

Software is fragile

Like all digital information, FOSS is fragile

- inconsiderate and/or malicious code loss (e.g., Code Spaces)
- business-driven code loss (e.g., Gitorious, Google Code)
- for obsolete code: physical media decay (data rot)

Where is the archive…

where we go if (a repository on) GitHub or GitLab.com goes away?

Software lacks its own research infrastructure

Photo: ALMA(ESO/NAOJ/NRAO), R. Hills

A wealth of software research on crucial issues…

- safety, security; test, verification, proof;
- software engineering, software evolution;
- big data, machine learning, empirical studies;

Software lacks its own research infrastructure

Photo: ALMA(ESO/NAOJ/NRAO), R. Hills

A wealth of software research on crucial issues…

- safety, security; test, verification, proof;
- software engineering, software evolution;
- big data, machine learning, empirical studies;

If you study the stars, you go to Atacama…

… where is the *very large telescope* of source code?

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Our mission

Collect, preserve and share the *source code* of *all the software* that is publicly available.

Past, present and future

Preserving the past, *enhancing* the present, *preparing* the future.

Our principles

Our principles

Open approach

- 100% FOSS
- transparency

In for the long haul

- replication
- non profit

Archiving goals

Targets: VCS repositories & source code releases (e.g., tarballs)

We DO archive

- file content (= blobs)
- revisions (= commits), with full metadata
- releases (= tags), ditto
- where (origin) & when (visit) we found any of the above

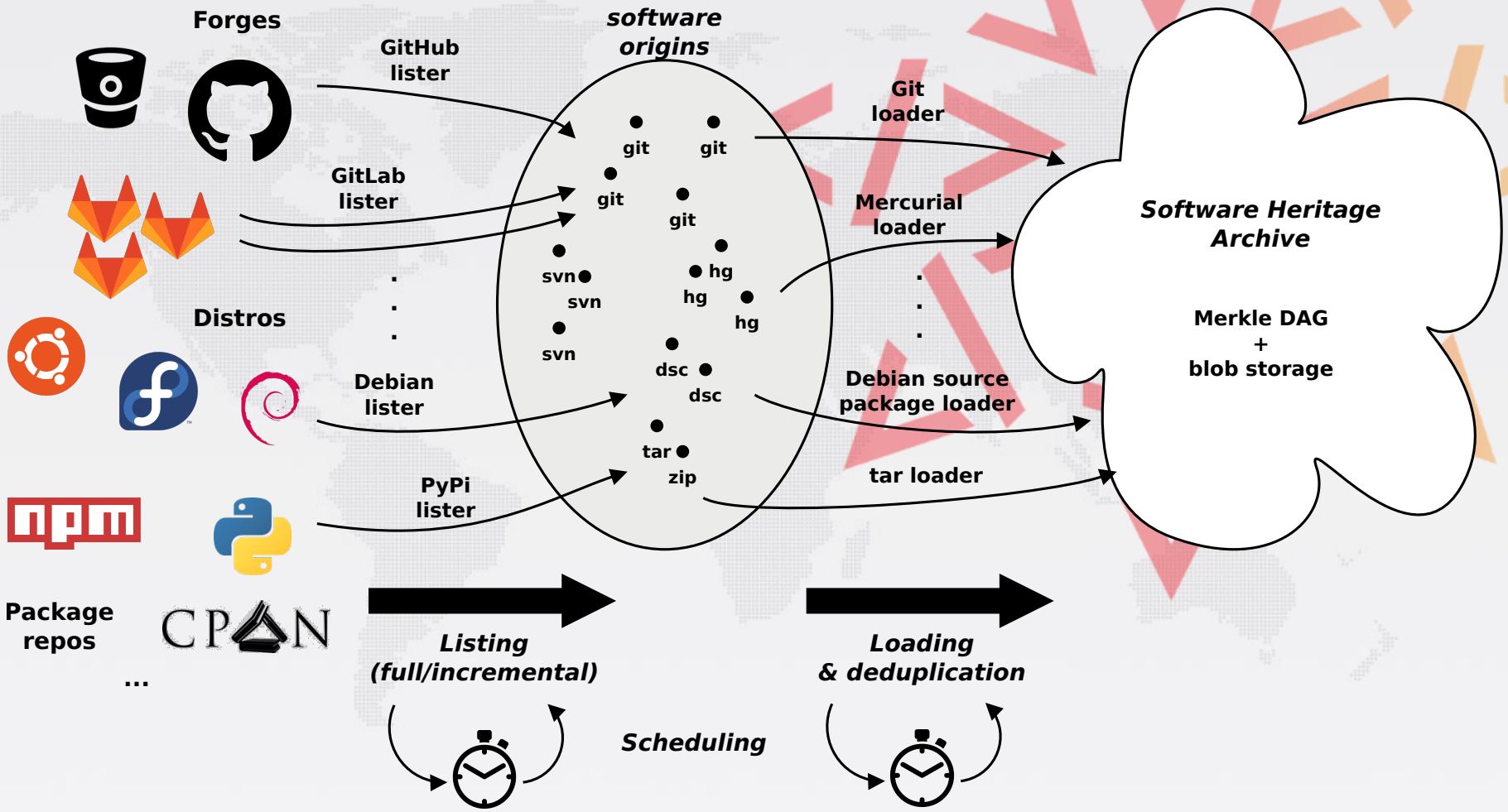
... in a VCS-/archive-agnostic canonical data model

We DON'T archive

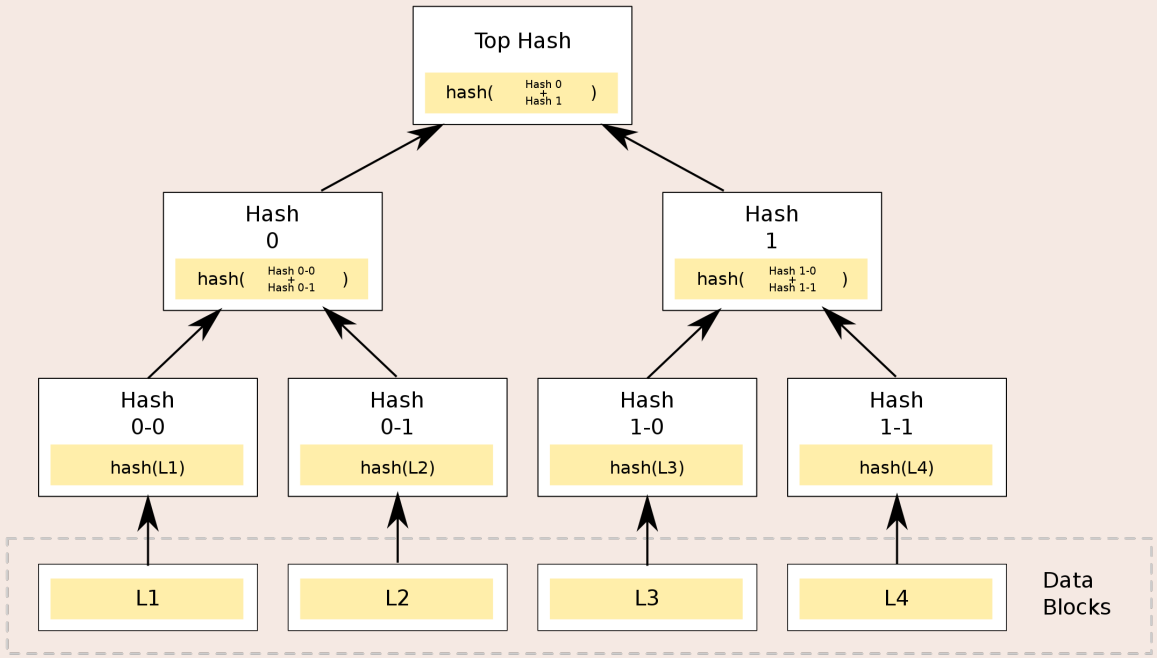
- homepages, wikis
- BTS/issues/code reviews/etc.
- mailing lists

Long term vision: play our part in a *"semantic wikipedia of software"*

Data flow


Forges
GitHub lister
GitLab lister
Distros
Debian lister
PyPi lister
Package repos
CPAN
...
software origins
git
git
git
git
svn
svn
svn
hg
hg
hg
dsc
dsc
tar
zip
Git loader
Mercurial loader
Debian source package loader
tar loader
Software Heritage Archive
Merkle DAG
+
blob storage
Listing (full/incremental)
Loading & deduplication
Scheduling

Merkle tree (R. C. Merkle, Crypto 1979)

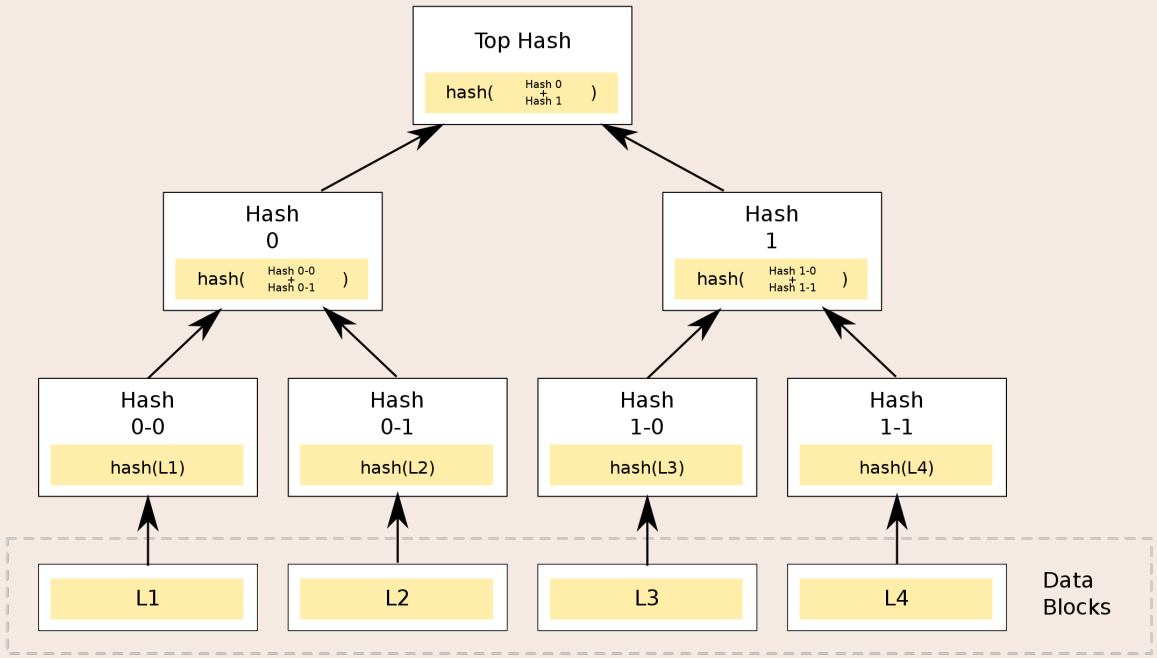


Combination of

- tree
- hash function

Merkle trees

Merkle tree (R. C. Merkle, Crypto 1979)



Combination of

- tree
- hash function

Classical cryptographic construction

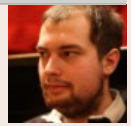
- fast, parallel signature of large data structures
- widely used (e.g., Git, blockchains, IPFS, …)
- built-in deduplication

Example: a Software Heritage revision

Revisions

Details | Changes | Files

SHA: 963634dca6ba5dc37e3ee426ba091092c267f9f6



Author: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep 1 14:26:13 2016)

Committer: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep 1 14:26:13 2016)

Subject: **provenance.tasks: add the revision -> origin cache task**

Parent: fc3a8b59ca1df424d860f2c29ab07fee4dc35d10 : *test_storage: properly pipeline origin and cont...*

provenance.tasks: add the revision -> origin cache task

swh/storage/provenance/tasks.py 77

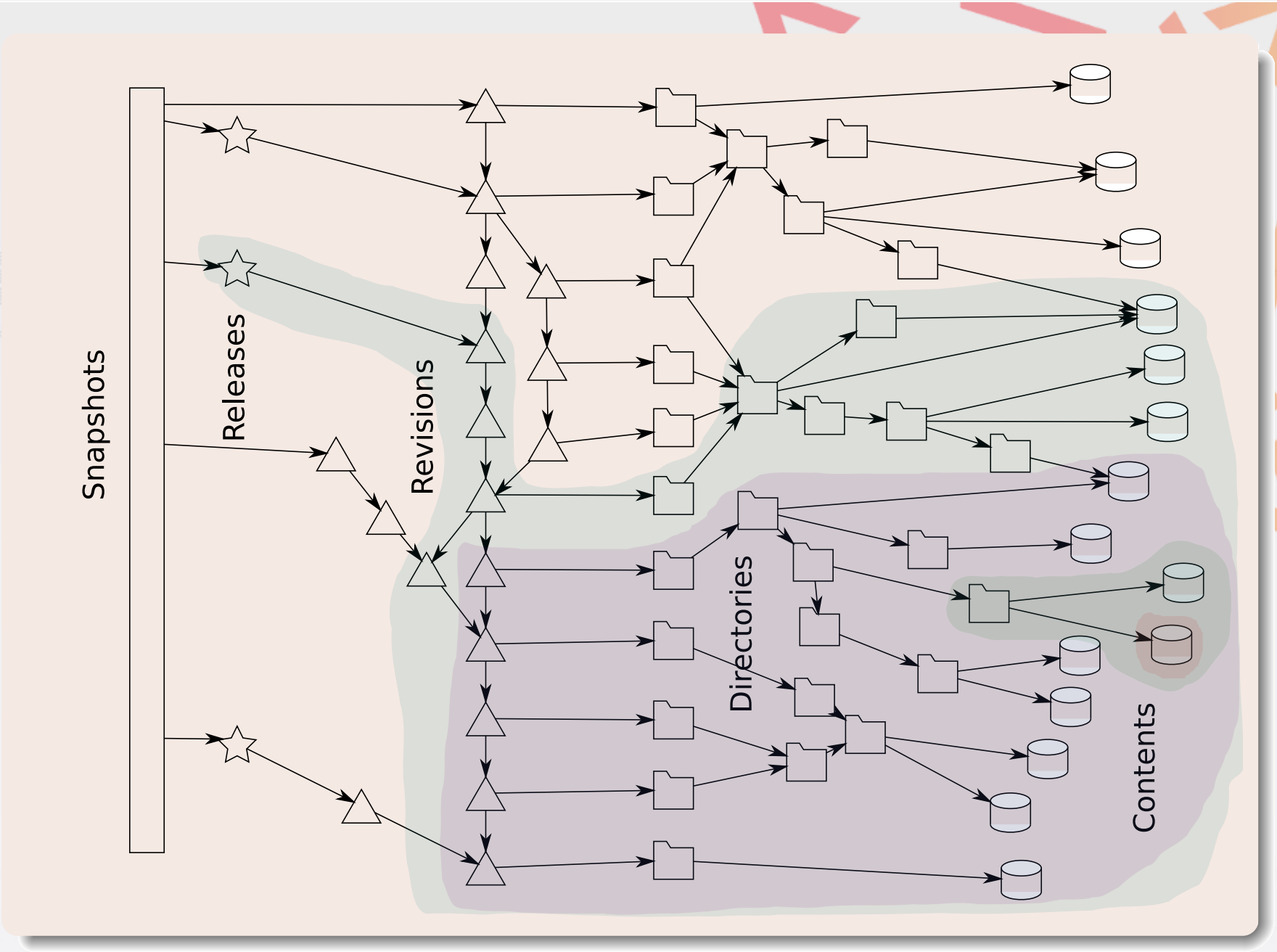
tree 515f00d44e92c65322aaa9bf3fa097c00ddb9c7d
parent fc3a8b59ca1df424d860f2c29ab07fee4dc35d10
author Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200
committer Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200

provenance.tasks: add the revision -> origin cache task

id: 963634dca6ba5dc37e3ee426ba091092c267f9f6

Note: most object kinds currently have Git-compatible identifiers

The archive: a (giant) Merkle DAG



Software stack

3rd party

- Debian, Puppet
- PostgreSQL for metadata storage, with barman & pglogical
- RabbitMQ for task scheduling
- Python3 and psycopg2 for the backend
- Flask and Bootstrap for Web stuff
- Phabricator

in house

- *ad hoc* object storage (to avoid imposing tech to mirros)
- data model implementation, listers, loaders, scheduler
- ~50 Git repositories (~20 Python packages, ~10 Puppet modules)
- ~30 kSLOC Python / ~12 kSLOC SQL / ~4 kSLOC Puppet
- licence choice: GPLv3 (backend) / AGPLv3 (frontend)

Hardware stack

in house

- 2x hypervisors with ~20 VMs
- 2x high density storage array (60 * 6TB => 300TB usable)

on Azure

- full object storage mirror (WIP)
- workers for content indexing

Archive coverage

Our sources

- GitHub — full, up-to-date mirror
- Debian — daily snapshots of all suites since 2005–2015
- GNU — all releases as of August 2015
- Gitorious, Google Code — local copy (Archive Team & Google)

Archive coverage

Our sources

- GitHub — full, up-to-date mirror
- Debian — daily snapshots of all suites since 2005–2015
- GNU — all releases as of August 2015
- Gitorious, Google Code — local copy (Archive Team & Google)

Some numbers

150 TB blobs, 6 TB database (as a graph: 5 B nodes + 50 B edges)

Archive coverage

Our sources

- GitHub — full, up-to-date mirror
- Debian — daily snapshots of all suites since 2005–2015
- GNU — all releases as of August 2015
- Gitorious, Google Code — local copy (Archive Team & Google)

Some numbers

150 TB blobs, 6 TB database (as a graph: 5 B nodes + 50 B edges)

The *richest* source code archive already, … and growing daily!

Web API

Fresh from the oven: first public version of our Web API

https://archive.softwareheritage.org/api/

Web API

Fresh from the oven: first public version of our Web API
https://archive.softwareheritage.org/api/

Features

- pointwise browsing of the Software Heritage archive
  - … releases ⟶ revisions ⟶ directories ⟶ contents …
- full access to the metadata of archived objects
- crawling information
  - *when have you last visited this Git repository I care about?*
  - *where were its branches/tags pointing to at the time?*

Web API

Fresh from the oven: first public version of our Web API

`https://archive.softwareheritage.org/api/`

Features

- pointwise browsing of the Software Heritage archive
  - … releases ⟶ revisions ⟶ directories ⟶ contents …
- full access to the metadata of archived objects
- crawling information
  - *when have you last visited this Git repository I care about?*
  - *where were its branches/tags pointing to at the time?*

Complete endpoint index

`https://archive.softwareheritage.org/api/1/`

A tour of the Web API — origins & visits

```
GET https://archive.softwareheritage.org/api/1/origin/ \
      git/url/https://github.com/hylang/hy
{ "id": 1,
  "origin_visits_url": "/api/1/origin/1/visits/",
  "type": "git",
  "url": "https://github.com/hylang/hy"
}
```

```
GET https://archive.softwareheritage.org/api/1/origin/ \
      1/visits/
[ ...,
  { "date": "2016-09-14T11:04:26.769266+00:00",
    "origin": 1,
    "origin_visit_url": "/api/1/origin/1/visit/13/",
    "status": "full",
    "visit": 13
  }, ...
]
```

A tour of the Web API — snapshots

```
GET https://archive.softwareheritage.org/api/1/origin/ \
      1/visit/13/
{ ...,
  "occurrences": { ...,
    "refs/heads/master": {
      "target": "b94211251...",
      "target_type": "revision",
      "target_url": "/api/1/revision/b94211251.../"
    },
    "refs/tags/0.10.0": {
      "target": "7045404f3...",
      "target_type": "release",
      "target_url": "/api/1/release/7045404f3.../"
    }, ...
  },
  "origin": 1,
  "origin_url": "/api/1/origin/1/",
  "status": "full",
  "visit": 13
}
```

A tour of the Web API — revisions

```
GET https://archive.softwareheritage.org/api/1/revision/ \
      6072557b6c10cd9a21145781e26ad1f978ed14b9/
{
  "author": {
    "email": "tag@pault.ag",
    "fullname": "Paul Tagliamonte <tag@pault.ag>",
    "id": 96,
    "name": "Paul Tagliamonte"
  },
  "committer": { ... },
  "date": "2014-04-10T23:01:11-04:00",
  "committer_date": "2014-04-10T23:01:11-04:00",
  "directory": "2df4cd84e...",
  "directory_url": "/api/1/directory/2df4cd84e.../",
  "history_url": "/api/1/revision/6072557b6.../log/",
  "merge": false,
  "message": "0.10: The Oh f*ck it's PyCon release",
  "parents": [ {
     "id": "10149f66e...",
     "url": "/api/1/revision/10149f66e.../"
  } ],
```

A tour of the Web API — contents

```
GET https://archive.softwareheritage.org/api/1/content/ \
      adc83b19e793491b1c6ea0fd8b46cd9f32e592fc/
{
  "data_url": "/api/1/content/sha1:adc83b19e.../raw/",
  "filetype_url": "/api/1/content/sha1:.../filetype/",
  "language_url": "/api/1/content/sha1:.../language/",
  "length": 1,
  "license_url": "/api/1/content/sha1:.../license/",
  "sha1": "adc83b19e...",
  "sha1_git": "8b1378917...",
  "sha256": "01ba4719c...",
  "status": "visible"
}
```

A tour of the Web API — contents

```
GET https://archive.softwareheritage.org/api/1/content/ \
      adc83b19e793491b1c6ea0fd8b46cd9f32e592fc/
{
  "data_url": "/api/1/content/sha1:adc83b19e.../raw/",
  "filetype_url": "/api/1/content/sha1:.../filetype/",
  "language_url": "/api/1/content/sha1:.../language/",
  "length": 1,
  "license_url": "/api/1/content/sha1:.../license/",
  "sha1": "adc83b19e...",
  "sha1_git": "8b1378917...",
  "sha256": "01ba4719c...",
  "status": "visible"
}
```

Caveats

- rate limits apply throughout the API
- blob download not available yet

Roadmap

Features...

- (done) lookup by content hash
- browsing: "wayback machine" for archived code
  - (done) via Web API
  - (todo) via Web UI
- (todo) download: `wget` / `git clone` from the archive
- (todo) provenance information for all archived content
- (todo) full-text search on all archived source code files

Roadmap

Features...

- (done) lookup by content hash
- browsing: "wayback machine" for archived code
  - (done) via Web API
  - (todo) via Web UI
- (todo) download: `wget` / `git clone` from the archive
- (todo) provenance information for all archived content
- (todo) full-text search on all archived source code files

... and much more than one could possibly imagine

all the world's software development history in a single graph!

Challenges — scaling

- big, but not *that* big — it's all text (in the good repos…)

Challenges — scaling

- big, but not *that* big — it's all text (in the good repos...)
- object storage
  - hundreds of TB is taxing for volunteer mirror operators
  - good replication properties: append only, self healing
  - costly extraordinary maintenance, e.g., primary key changes

Challenges — scaling

- big, but not *that* big — it's all text (in the good repos…)
- object storage
  - hundreds of TB is taxing for volunteer mirror operators
  - good replication properties: append only, self healing
  - costly extraordinary maintenance, e.g., primary key changes
- Merkle DAG
  - good choice to counter hosting site inflation
  - beyond the state of the art of graph databases (?)
  - e.g., provenance queries are expensive
  - mitigation: (large) caches

Challenges — scaling

- big, but not *that* big — it's all text (in the good repos…)
- object storage
  - hundreds of TB is taxing for volunteer mirror operators
  - good replication properties: append only, self healing
  - costly extraordinary maintenance, e.g., primary key changes
- Merkle DAG
  - good choice to counter hosting site inflation
  - beyond the state of the art of graph databases (?)
  - e.g., provenance queries are expensive
  - mitigation: (large) caches
- full text indexes & friends
  - might be arbitrary large, but entirely derived data

Challenges — operational transparency

- the mission is more important than any of us
- how can we *prove* we're pursuing it as soundly as possible?
- … and *recover* from mistakes if/when they happen?

Challenges — operational transparency

- the mission is more important than any of us
- how can we *prove* we're pursuing it as soundly as possible?
- … and *recover* from mistakes if/when they happen?

- difficult at this scale
- some elements of response:
  - 100% FOSS
  - non-profit endeavor
  - full, public ledger of all changes to all data throughout their entire life cycle — ingestion/maintenance/mirroring/… (?)

You can help!

Coding

- `www.softwareheritage.org/community/developers/`
- `forge.softwareheritage.org` — our own code

Current development priorities

★★★	listers for unsupported forges, distros, pkg. managers
★★★	loaders for unsupported VCS, source package formats
★★	Web UI: eye candy wrapper around the Web API
★	content indexing and search

… *all* contributions equally welcome!

You can help!

Coding

- `www.softwareheritage.org/community/developers/`
- `forge.softwareheritage.org` — our own code

Current development priorities

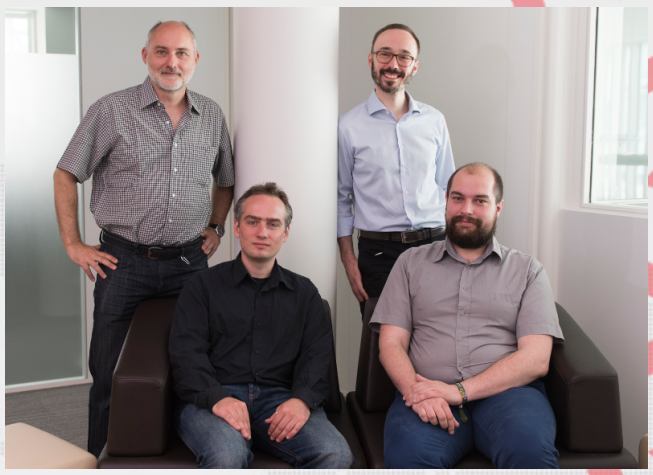
★★★	listers for unsupported forges, distros, pkg. managers
★★★	loaders for unsupported VCS, source package formats
★★	Web UI: eye candy wrapper around the Web API
★	content indexing and search

... *all* contributions equally welcome!

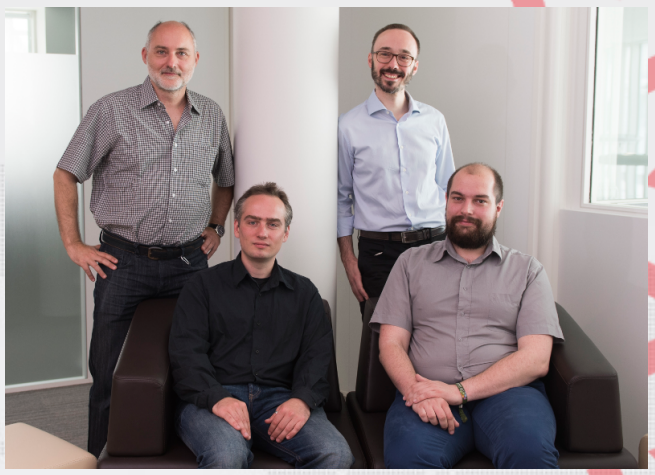
Join us

- `www.softwareheritage.org/jobs` — job openings
- `wiki.softwareheritage.org` — internships

The Software Heritage community



The Software Heritage community

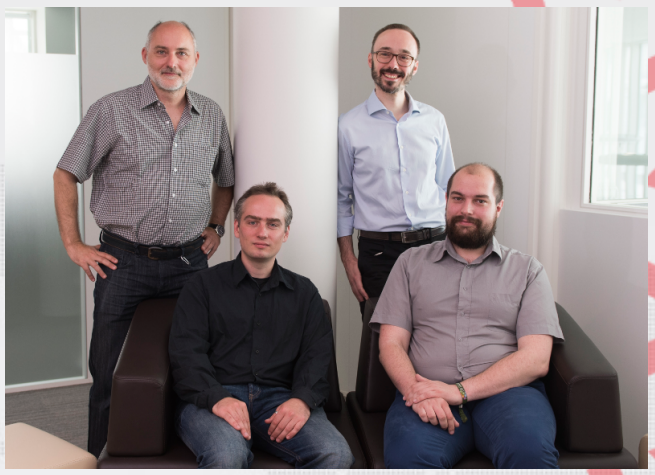


Inria as initiator

Inria
INVENTEURS DU MONDE NUMÉRIQUE

- .fr national CS research institution
- strong FOSS culture, W3C founding partner

The Software Heritage community



Inria as initiator

- .fr national CS research institution
- strong FOSS culture, W3C founding partner

Early Partners and Supporters

Société Générale, Microsoft, Huawei, Nokia Bell Labs, DANS, ACM, Creative Commons, Eclipse, Engineering, FSF, Gandi, GitHub, IEEE, OIN, OSI, OW2, Software Freedom Conservancy, SFLC, The Document Foundation, The Linux Foundation, ...

Conclusion

Software Heritage is

- a *reference archive* of *all* FOSS ever written
- a unique *complement* for *development platforms*
- an international, open, nonprofit, *mutualized infrastructure*
- at the service of our community, at the service of society

Come in, we're open!

`www.softwareheritage.org` — *sponsoring, job openings*
`wiki.softwareheritage.org` — *internships, leads*
`forge.softwareheritage.org` — *our own code*

Questions?

Q: how about SHA1 collisions?

```
create domain sha1 as bytea
  check (length(value) = 20);
create domain sha1_git as bytea
  check (length(value) = 20);
create domain sha256 as bytea
  check (length(value) = 32);

create table content (
  sha1       sha1 primary key,
  sha1_git   sha1_git not null,
  sha256     sha256 not null,
  length     bigint not null,
  ctime      timestamptz not null default now(),
  status     content_status not null default 'visible',
  object_id  bigserial
);

create unique index on content(sha1_git);
create unique index on content(sha256);
```

Q: do you archive *only* Free Software?

- We only crawl origins *meant* to host source code (e.g., forges)
- Most (~90%) of what we *actually* retrieve is textual content

- Our goal: archive *the entire Free Software commons*

- Large parts of what we retrieve is *already* Free Software, today
- Most of the rest *will become* Free Software in the long term
  - e.g., at copyright expiration