Preserving Source Code

Challenges and Opportunities for the Reproductibility of Science

Stefano Zacchiroli

University Paris Diderot & Inria
zack@upsilon.cc

25 May 2017
DAUIN, Politecnico di Torino — Turin, Italy

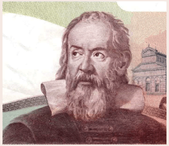
Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Outline

1. Reproducibility
2. Software Heritage
3. Technical overview
4. Current status
5. Outlook

How we built our scientific knowledge

The experimental method

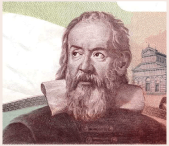


- make an *observation*
- formulate an *hypothesis*
- set up an experiment
- formulate a *theory*

And then we reproduce and verify.

How we built our scientific knowledge

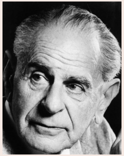
The experimental method



- make an *observation*
- formulate an *hypothesis*
- set up an experiment
- formulate a *theory*

And then we reproduce and verify.

Reproducibility is the key



> *non-reproducible single occurrences are of no significance to science*
>
> *Karl Popper, The Logic of Scientific Discovery, 1934*

Reproducibility, today

Reproducibility (Wikipedia)

the ability of an entire experiment or study to be *reproduced*, either by the researcher or *by someone else working independently*.
It is one of the main principles of the scientific method.

Why we want it

- foundation of the scientific method
- accelerator of research: allows to build upon previous work
- visibility: reproducible results are cited more often
- transparency of results eases acceptance
- necessary for industrial transfer

reproducibility is *the essence* of *industry*!

Reproducibility in the digital age

For an experiment involving software, we need

open access to the scientific article describing it

open data sets used in the experiment

source code of all the components

environment of execution

stable references between all this

Reproducibility in the digital age

For an experiment involving software, we need

open access to the scientific article describing it

open data sets used in the experiment

source code of all the components

environment of execution

stable references between all this

Remark

The first two items are already widely discussed!

. . . what about *software*?

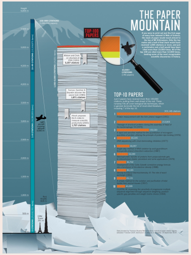
Software is Knowledge

Software is *an essential component* of modern scientific research

Top 100 papers (Nature, October 2014)

> *[. . .] the vast majority describe experimental methods or software that have become essential in their fields.*

`http://www.nature.com/news/the-top-100-papers-1.16224`



Software and reproducibility

A fundamental question

How are we doing, regarding reproducibility, in *Software*?

The case of Computer Systems Research

A field with Computer experts … we have high expectations!
Christian Collberg set out to check them.

Measuring Reproducibility in Computer Systems Research

Long and detailed technical report, March 2014
`http://reproducibility.cs.arizona.edu/v1/tr.pdf`

Collberg's report from the trenches

Analysis of 613 papers

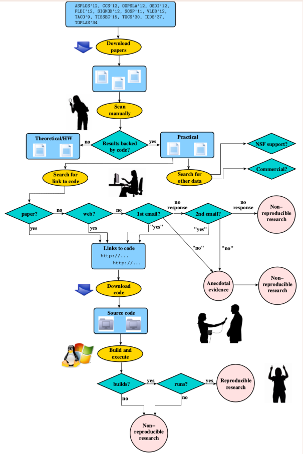
- 8 ACM conferences: ASPLOS'12, CCS'12, OOPSLA'12, OSDI'12, PLDI'12, SIGMOD'12, SOSP'11, VLDB'12
- 5 journals: TACO'9, TISSEC'15, TOCS'30, TODS'37, TOPLAS'34

all very practical oriented

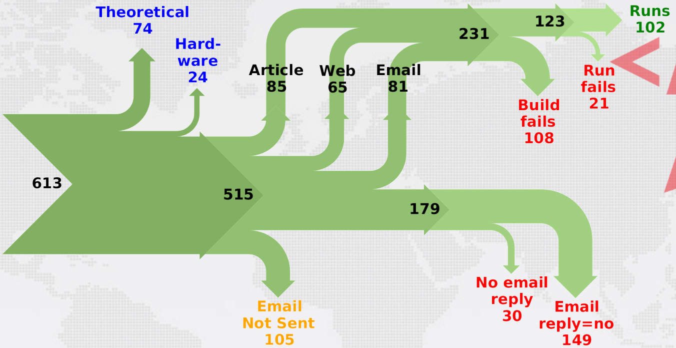
The basic question

can we get the code to build and run?

The workflow



The result



This can be debated (see http://cs.brown.edu/~sk/Memos/Examining-Reproducibility/), but...

... that's a whopping 81% of non reproducible works!

The reasons (or, “the dog ate my program”)

Why so much software fails to pass the test?

Many issues, nice anecdotes, and it finally boils down to

- *Availability*
- *Traceability*
- Environment
- Automation (do *you* use continuous integration?)
- Documentation
- Understanding (including free/open source software)

The reasons (or, “the dog ate my program”)

Why so much software fails to pass the test?

Many issues, nice anecdotes, and it finally boils down to

- *Availability*
- *Traceability*
- Environment
- Automation (do *you* use continuous integration?)
- Documentation
- Understanding (including free/open source software)

The first two are important *software preservation issues*

Yes, code is fragile:

it can be destroyed, and we can lose trace of it

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?

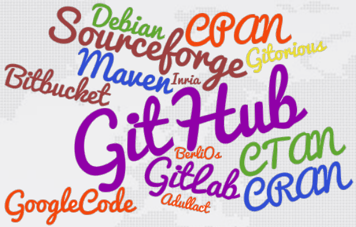
Software is spread all around

Sourceforge
Debian
CPAN
Gitorious
Maven
Inria
Bitbucket
GitHub
BerliOs
CTAN
Gitlab
CRAN
GoogleCode
Adullact

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?

Outline

1. Reproducibility
2. Software Heritage
3. Technical overview
4. Current status
5. Outlook

The Software Heritage Project

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Our mission

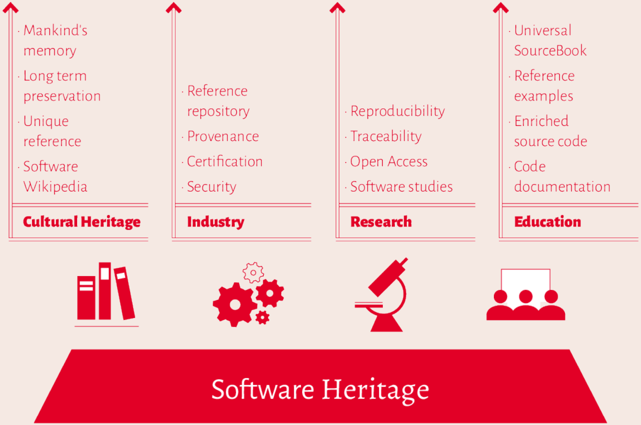
Collect, preserve and share the *source code* of *all the software* that is publicly available.

Past, present and future

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

We are working on the foundations

one infrastructure to build them all



- Mankind's memory
- Long term preservation
- Unique reference
- Software Wikipedia

Cultural Heritage

- Reference repository
- Provenance
- Certification
- Security

Industry

- Reproducibility
- Traceability
- Open Access
- Software studies

Research

- Universal SourceBook
- Reference examples
- Enriched source code
- Code documentation

Education

Software Heritage

Supporting more accessible and reproducible science

A global library referencing all software used in all research fields

- completes the infrastructure for Open Access in science
- provides intrinsic persistent identifiers needed for scientific reproducibility
- enables large scale, verifiable software studies

Software lacks its own research infrastructure

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

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?

Better software for industry and society

A unique reference catalog of all industrial software components

- a single entry point to discover, explore and reuse source code
- eases vulnerability tracking for more secure software
- simplifies traceability for better software integration
- ensures long term preservation of critical software

Fostering wider education to computing

A global source referencing all software

- a source book for technological education
- intrinsic persistent identifiers for stable course materials
- enables real-world, semi-automated documentation

Outline

1 Reproducibility

2 Software Heritage

3 Technical overview

4 Current status

5 Outlook

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

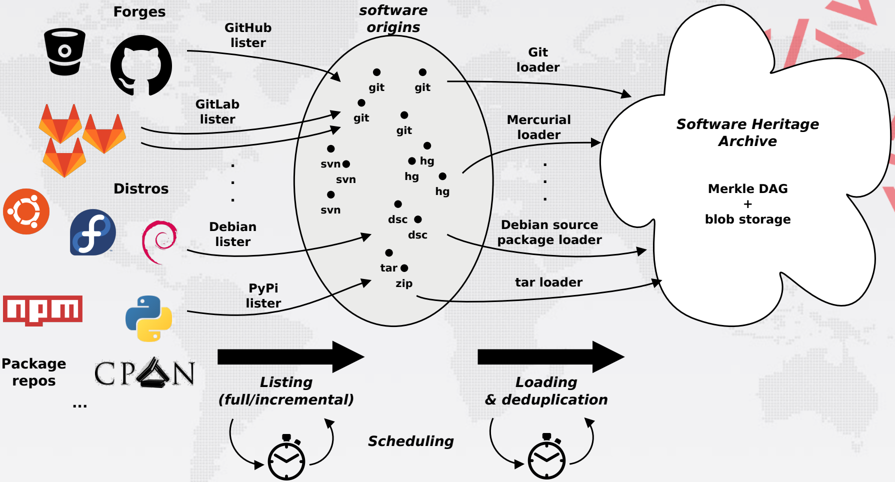
… in a VCS-/archive-agnostic canonical data model

We DON'T archive

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

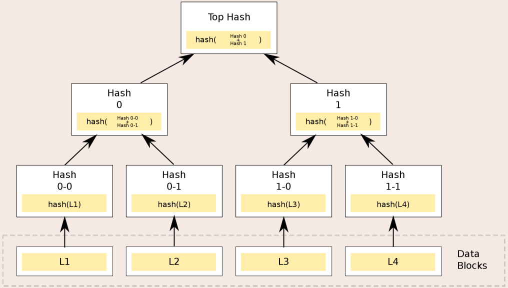
Long term vision: play our part in a *"semantic wikipedia of software"*

Data flow


Forges
GitHub lister
GitLab lister
Distros
Debian lister
PyPi lister
Package repos
...
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 trees

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

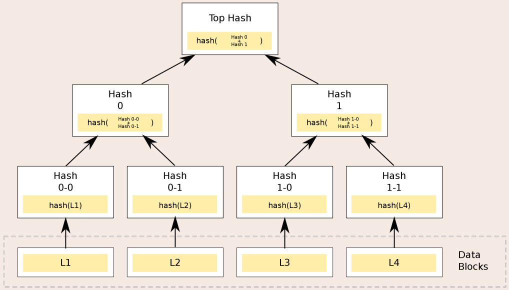


Combination of

- tree
- hash function

Merkle trees

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



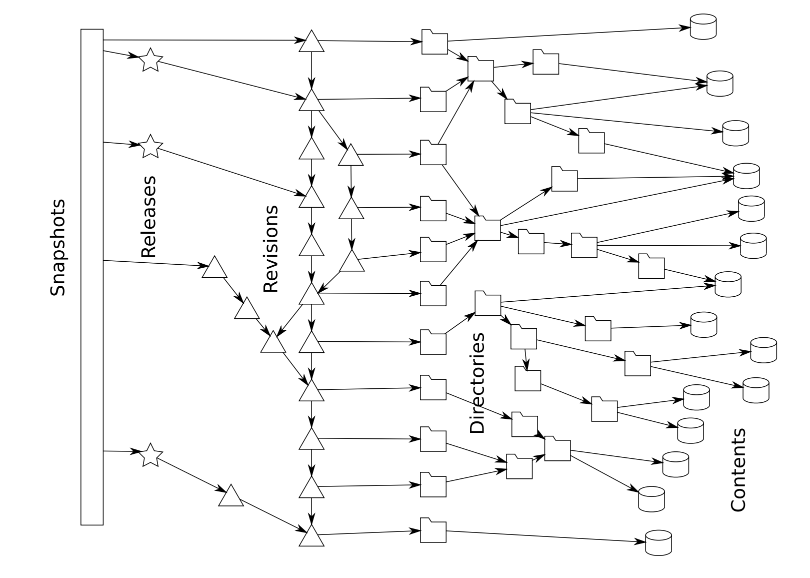
Combination of

- tree
- hash function

Classical cryptographic construction

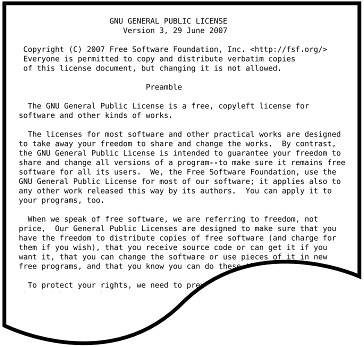
- fast, parallel signature of large data structures
- widely used (e.g., Git, blockchains, IPFS, ...)
- built-in deduplication

The Software Heritage archive: a gigantic Merkle DAG



The Software Heritage archive: a gigantic Merkle DAG

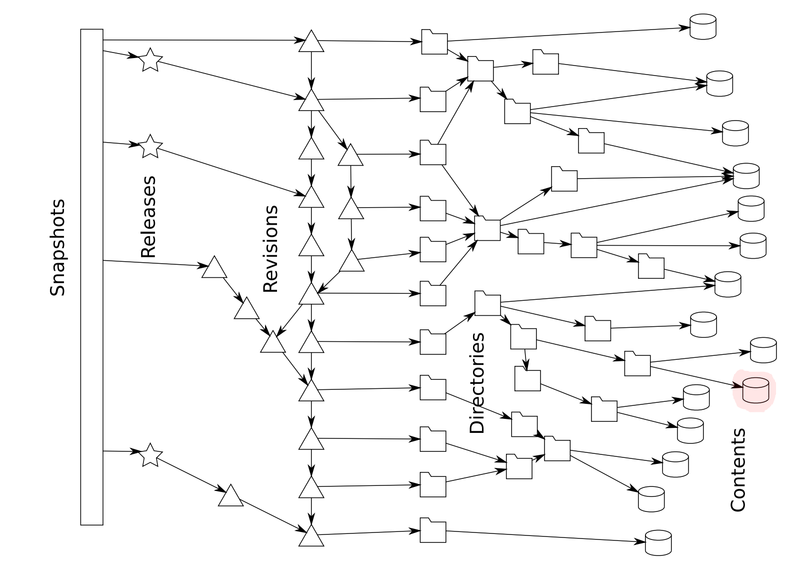
Contents



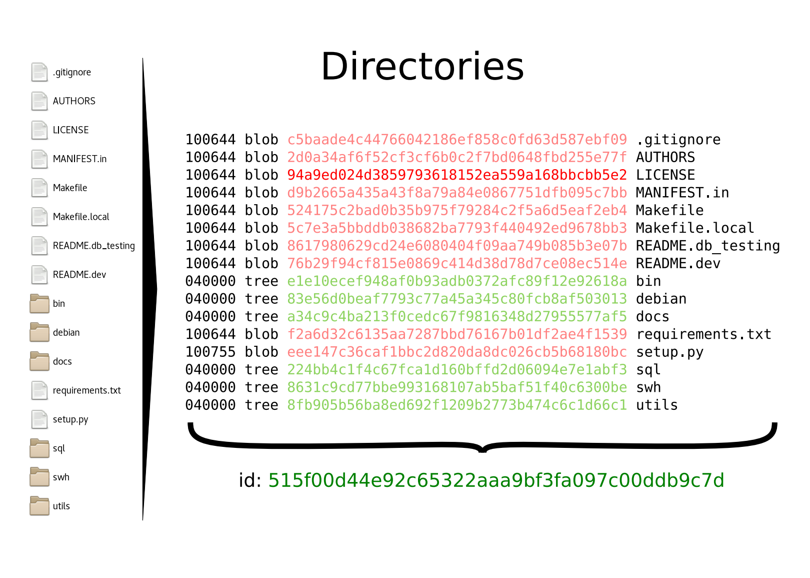
```
                    GNU GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The GNU General Public License is a free, copyleft license for
software and other kinds of works.

  The licenses for most software and other practical works are designed
to take away your freedom to share and change the works.  By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.  We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors.  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these

  To protect your rights, we need to pre
```

sha1: 8624bcdae55baeef...
sha256: 8ceb4b9ee5aded...
sha1_git: 94a9ed024d385...
length: 35147

The Software Heritage archive: a gigantic Merkle DAG



The Software Heritage archive: a gigantic Merkle DAG

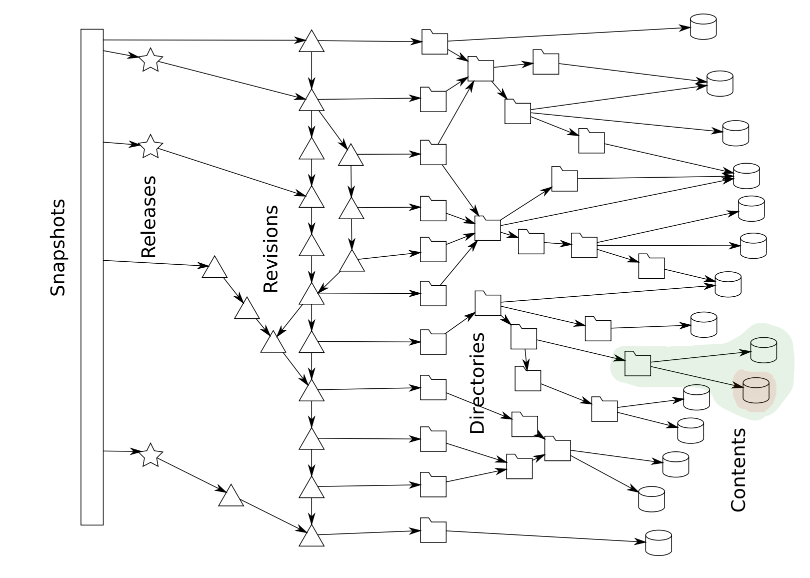


Directories

```
100644 blob c5baade4c44766042186ef858c0fd63d587ebf09 .gitignore
100644 blob 2d0a34af6f52cf3cf6b0c2f7bd0648fbd255e77f AUTHORS
100644 blob 94a9ed024d3859793618152ea559a168bbcbb5e2 LICENSE
100644 blob d9b2665a435a43f8a79a84e0867751dfb095c7bb MANIFEST.in
100644 blob 524175c2bad0b35b975f79284c2f5a6d5eaf2eb4 Makefile
100644 blob 5c7e3a5bbddb038682ba7793f440492ed9678bb3 Makefile.local
100644 blob 8617980629cd24e6080404f09aa749b085b3e07b README.db_testing
100644 blob 76b29f94cf815e0869c414d38d78d7ce08ec514e README.dev
040000 tree e1e10ecef948af0b93adb0372afc89f12e92618a bin
040000 tree 83e56d0beaf7793c77a45a345c80fcb8af503013 debian
040000 tree a34c9c4ba213f0cedc67f9816348d27955577af5 docs
100644 blob f2a6d32c6135aa7287bbd76167b01df2ae4f1539 requirements.txt
100755 blob eee147c36caf1bbc2d820da8dc026cb5b68180bc setup.py
040000 tree 224bb4c1f4c67fca1d160bffd2d06094e7e1abf3 sql
040000 tree 8631c9cd77bbe993168107ab5baf51f40c6300be swh
040000 tree 8fb905b56ba8ed692f1209b2773b474c6c1d66c1 utils
```

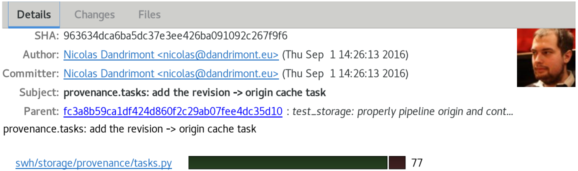
id: 515f00d44e92c65322aaa9bf3fa097c00ddb9c7d

The Software Heritage archive: a gigantic Merkle DAG



The Software Heritage archive: a gigantic Merkle DAG

Revisions



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...*

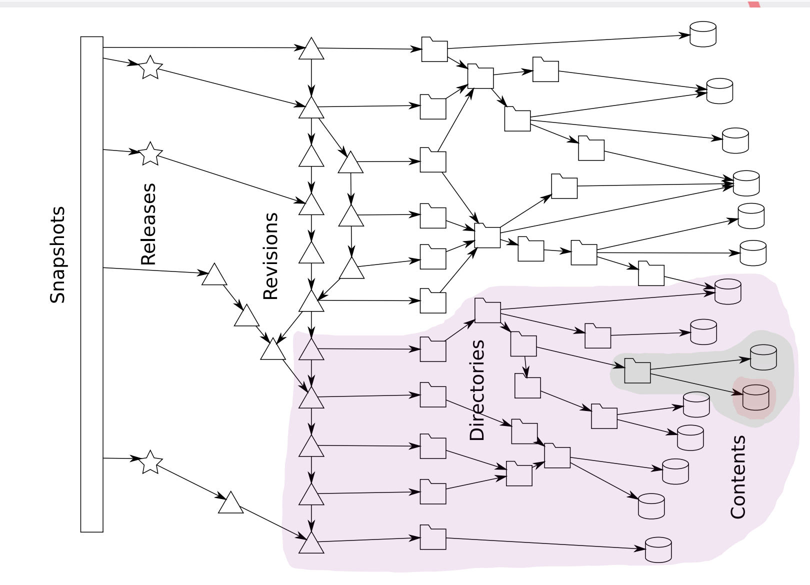
provenance.tasks: add the revision -> origin cache task

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
```

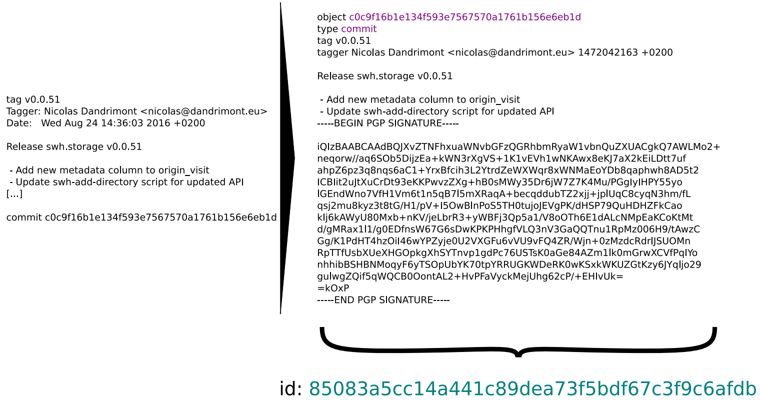
id: 963634dca6ba5dc37e3ee426ba091092c267f9f6

The Software Heritage archive: a gigantic Merkle DAG



The Software Heritage archive: a gigantic Merkle DAG

Releases

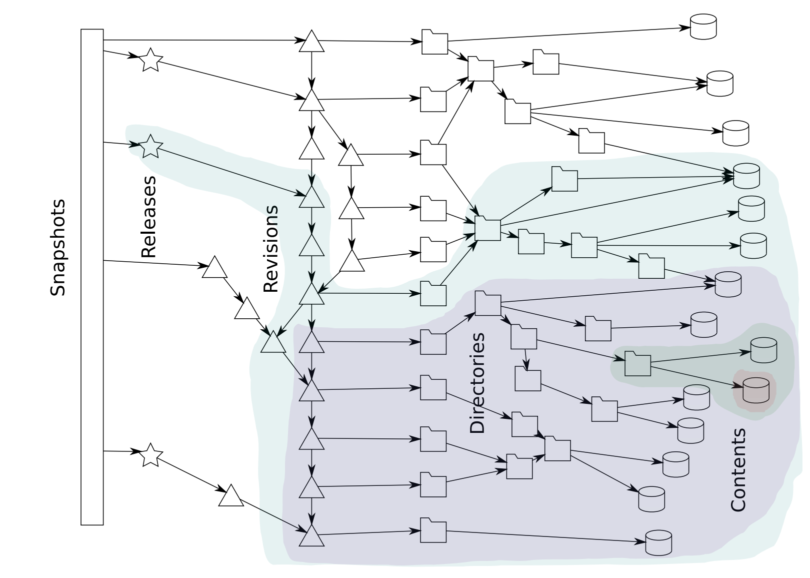


```
tag v0.0.51
Tagger: Nicolas Dandrimont <nicolas@dandrimont.eu>
Date:   Wed Aug 24 14:36:03 2016 +0200

Release swh.storage v0.0.51

 - Add new metadata column to origin_visit
 - Update swh-add-directory script for updated API
[...]

commit c0c9f16b1e134f593e7567570a1761b156e6eb1d
```

```
object c0c9f16b1e134f593e7567570a1761b156e6eb1d
type commit
tag v0.0.51
tagger Nicolas Dandrimont <nicolas@dandrimont.eu> 1472042163 +0200

Release swh.storage v0.0.51

 - Add new metadata column to origin_visit
 - Update swh-add-directory script for updated API
-----BEGIN PGP SIGNATURE-----

iQIzBAABCAAdBQJXvZTNFhxuaWNvbGFzQGRhbmRyaW1vbnQuZXUACgkQ7AWLMo2+
neqorw//aq6SOb5DijzEa+kWN3rXgVS+1K1vEVh1wNKAwx8eKJ7aX2kEiLDtt7uf
ahpZ6pz3q8nqs6aC1+YrxBfcih3L2YtrdZeWXWqr8xWNMaEoYDb8qaphwh8AD5t2
ICBIit2uJtXuCrDt93eKKPwvzZXg+hB0sMWy35Dr6jW7Z7K4Mu/PGgIyIHPY55yo
lGEndWno7VfH1Vm6t1n5qB7l5mXRaqA+becqddubTZ2xjj+jplUqC8cyqN3hm/fL
qsj2mu8kyz3t8tG/H1/pV+I5OwBlnPoS5TH0tujoJEVgPK/dHSP79QuHDHZFkCao
kIj6kAWyU80Mxb+nKV/jeLbrR3+yWBFj3Qp5a1/V8oOTh6E1dALcNMpEaKCoKtMt
d/gMRax1l1/g0EDfnsW67G6sDwKPKPHhgfVLQ3nV3GaQQTnu1RpMz006H9/tAwzC
Gg/K1PdHT4hzOiI46wYPZyje0U2VXGFu6vVU9vFQ4ZR/Wjn+0zMzdcRdrlJSUOMn
RpTTfUsbXUeXHGOpkgXhSYTnvp1gdPc76USTsK0aGe84AZm1lk0mGrwXCVfPqIYo
nhhibBSHBNMoqyF6yTSOpUbYK70tpYRRUGKWDeRK0wKSxkWKUZGtKzy6JYqljo29
gulwgZQif5qWQCB0OontAL2+HvPFaVyckMejUhg62cP/+EHIvUk=
=kOxP
-----END PGP SIGNATURE-----
```

id: 85083a5cc14a441c89dea73f5bdf67c3f9c6afdb

The Software Heritage archive: a gigantic Merkle DAG



Outline

1. Reproducibility
2. Software Heritage
3. Technical overview
4. Current status
5. Outlook

Archive coverage

Our sources

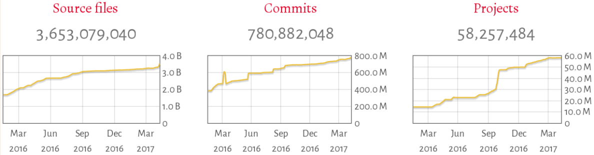
- GitHub — full, up-to-date mirror
- Debian, GNU — one shot ingestion experiment (up to Aug 2015)
- Gitorious, Google Code — local copy (Archive Team & Google)
- Bitbucket — WIP

Archive coverage

Our sources

- GitHub — full, up-to-date mirror
- Debian, GNU — one shot ingestion experiment (up to Aug 2015)
- Gitorious, Google Code — local copy (Archive Team & Google)
- Bitbucket — WIP

Some numbers



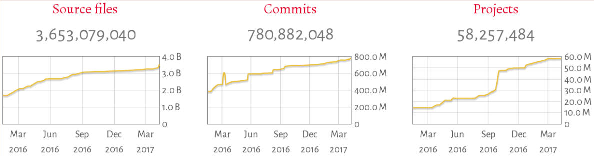
150 TB blobs, 5 TB database (as a graph: 7 B nodes + 60 B edges)

Archive coverage

Our sources

- GitHub — full, up-to-date mirror
- Debian, GNU — one shot ingestion experiment (up to Aug 2015)
- Gitorious, Google Code — local copy (Archive Team & Google)
- Bitbucket — WIP

Some numbers



150 TB blobs, 5 TB database (as a graph: 7 B nodes + 60 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 available for selected contents

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
  - might be arbitrary large, but entirely derived data
  - AST-based search won't work: too much diversity
  - "stupid" stemming? trigrams?

Challenges — operational accountability

- 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 accountability

- 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 & open development
  - full, public ledger of all changes to all data throughout their entire life cycle — ingestion/maintenance/mirroring/… (?)

Outline

1. Reproducibility
2. Software Heritage
3. Technical overview
4. Current status
5. Outlook

You can help… coders!

Coding

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

Join us

- www.softwareheritage.org/jobs — job openings
- wiki.softwareheritage.org/index.php?title=Internships — internships

You can help… scientists!

Community

- www.softwareheritage.org/community/scientists/
- swh-science@inria.fr
- wiki.softwareheritage.org/index.php?title=Working_groups

Working groups (planned)

- Extending the archive
  - Source Discovery and Ingestion
  - Metadata and Linked Data
- Evolving the archive
  - Modeling and Ingesting Version control systems
  - Distribution, Replication and Query

Working groups (planned)

- Connecting the archive
  - Reproducibility of Software
  - Open Access and Data
- Using the archive
  - Scientific API
  - Ethical and Legal Issues and Environment

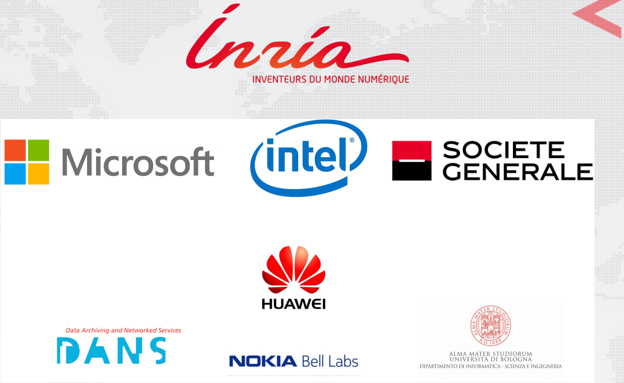
Sharing the Software Heritage vision

IEEE Software
GitLab
eclipse
OW2
fsfe
nlnet FOUNDATION
SIGSOFT
SPECIAL INTEREST GROUP ON SOFTWARE ENGINEERING
ENGINEERING
THE LINUX FOUNDATION
TDF
ADULLACT
INFORMATICS EUROPE
GitHub
software freedom conservancy
frdm
Software Freedom Law Center
Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
FREE SOFTWARE FOUNDATION
creative commons
AdaCore
The GNAT Pro Company
open source initiative
openinventionnetwork

See more

http://www.softwareheritage.org/support/testimonials

Sponsoring Software Heritage work


Inria
INVENTEURS DU MONDE NUMÉRIQUE
Microsoft
intel
SOCIETE GENERALE
HUAWEI
Data Archiving and Networked Services
DANS
NOKIA Bell Labs
ALMA MATER STUDIORUM
A.D. 1088
ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA
DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA

Going global

April 3rd, 2017: landmark UNESCO/Inria agreement. . .

www.softwareheritage.org/?p=11623

Next step: 27-28 Sep 2017: UNESCO/Inria conference in Paris

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, working groups*
`forge.softwareheritage.org` — *our own code*

References

Di Cosmo and Zacchiroli, *Software Heritage: How and Why to Preserve Software Source Code*, iPRES 2017 (to appear). Draft:
`https://upsilon.cc/~zack/stuff/software-heritage-draft.pdf`

FAQ: 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);
```