Software Heritage

Scholarly and Educational Synergies with Preserving our Software Commons

Stefano Zacchiroli

University Paris Diderot & Inria — zack@upsilon.cc

5 July 2017

22nd Annual Conf. on Innovation and Technology in Computer Science Education

Bologna, Italy

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

About the speaker

- `cs.unibo.it` alumni
- Computer Science researcher @ Univ. Paris Diderot & Inria
  - logics ⟶ applied formal methods ⟶ Free/Open Source Software (FOSS) engineering ⟶ digital preservation
- Computer Science teacher
  - from K-12 to graduate school students, I've always enjoyed my teaching "duties"!
  - current classes: software engineering (CS undergrad), FOSS (CS master)
- FOSS activist
  - Debian, Open Source Initiative, Free Software Foundation

Outline

Outline

Software is everywhere

Source code matters!

“The source code for a work means the preferred form of the work for making modifications to it."

GPL Licence

Source code matters!

“The source code for a work means the preferred form of the work for making modifications to it."

GPL Licence

Hello World

Source code matters!

"The source code for a work means the preferred form of the work for making modifications to it."

GPL Licence

Hello World

Program (excerpt of binary)

```
4004e6: 55
4004e7: 48 89 e5
4004ea: bf 84 05 40 00
4004ef: b8 00 00 00 00
4004f4: e8 c7 fe ff ff
4004f9: 90
4004fa: 5d
4004fb: c3
```

Source code matters!

"The source code for a work means the preferred form of the work for making modifications to it."

GPL Licence

Hello World

Program (excerpt of binary)

```
4004e6: 55
4004e7: 48 89 e5
4004ea: bf 84 05 40 00
4004ef: b8 00 00 00 00
4004f4: e8 c7 fe ff ff
4004f9: 90
4004fa: 5d
4004fb: c3
```

Program (source code)

```
/* Hello World program */

#include<stdio.h>

void main()
{
    printf("Hello World");
}
```

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

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?

Outline

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

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

Industry
- Reference repository
- Provenance
- Certification
- Security

Research
- Reproducibility
- Traceability
- Open Access
- Software studies

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




Software Heritage

Preserving the world's software heritage

A structured archive of all of the world's software

- preserve humanity's technological and scientific knowledge
- enable continued access to all digital documents and information
- building block for thematic portals and collections

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

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.

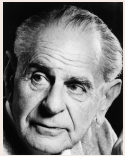
How we built our scientific knowledge

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

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

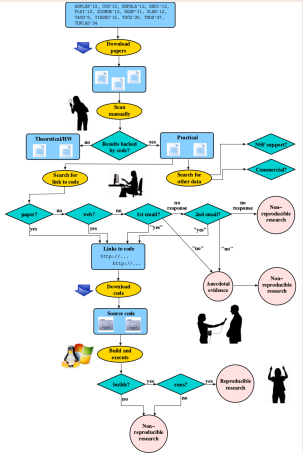
- 8 ACM conferences: ASPLOS'12, CCS'12, OOPSLA'12, OSDI'12, PLDI'12, SIGMOD'12, SOSP'11, VLDB'12
- 5 journals: TACO'9, TISSEC'15, TOCS'30, TODS'37, TOPLAS'34

all very practical oriented

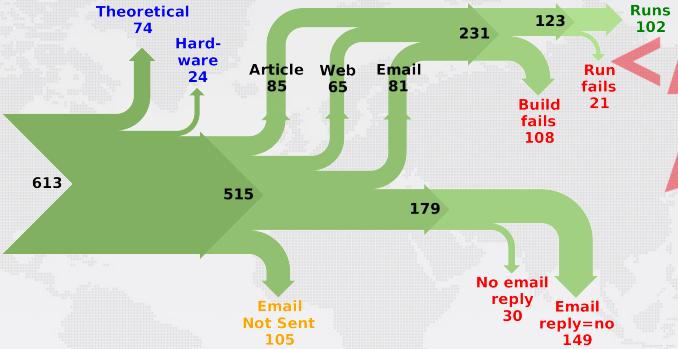
The basic question

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

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

Improving resources for computing education

Source books are popular in other fields, but still scarcely used in CS education.

The ultimate computing source book

- perfect basis for curating source books for all computing-related classes
  - relate pseudo-code/data structure/techniques to real-world implementations
  - follow implementation evolution through history
  - access historical contextual metadata (commits, timestamps, etc.)
  - assess impact, adoption, etc.
- intrinsic persistent identifiers and tracking for source code course materials

Collaborative curation wanted

- comprehensive software commons archive $\neq$ source code source book
- curation by motivated educators is needed to close the gap
- Software Heritage can offer perennity and the collaboration infrastructure …
- … as this ties very well into the "semantic wikipedia of software" vision
- who's up for the remaining challenge?

Our principles

Our principles

Open approach

- 100% FOSS
- transparency

In for the long haul

- replication
- non profit

Outline

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

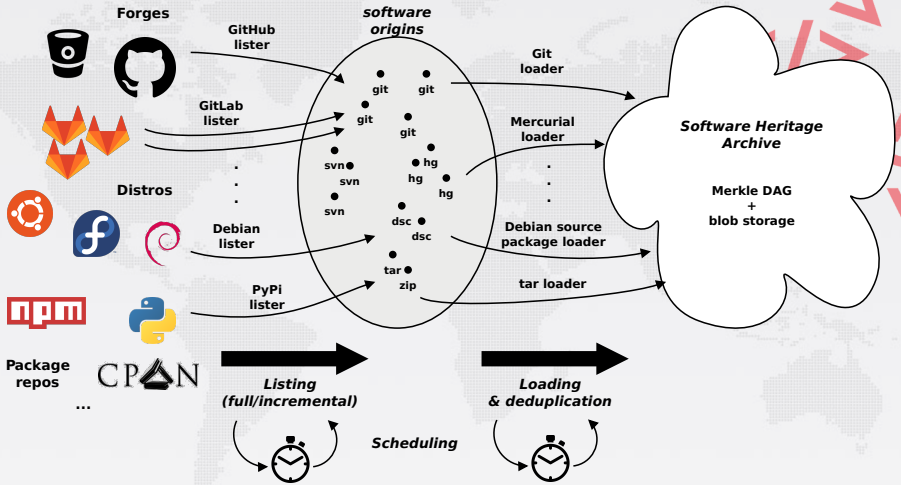
… in a VCS-/archive-agnostic canonical data model

We DON'T archive

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

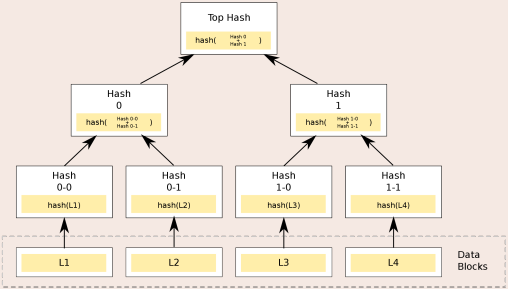
Long term vision: play our part in a *"semantic wikipedia of software"*

Data flow


Forges
GitHub lister
GitLab lister
Distros
Debian lister
PyPi lister
Package repos
...
software origins
git
svn
hg
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)

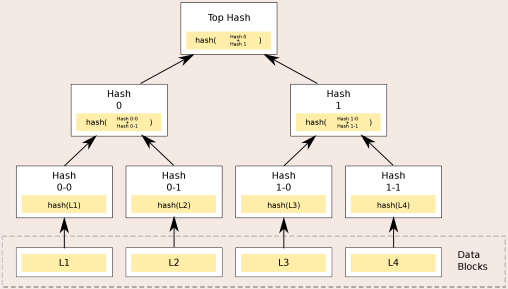


Combination of

- 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

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

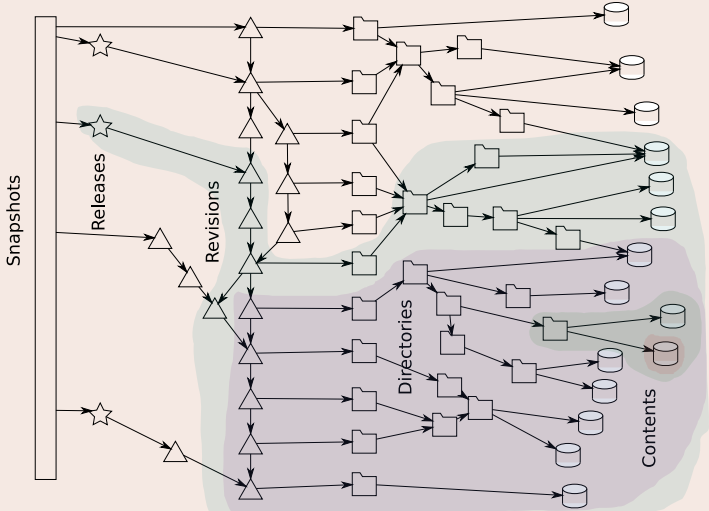
swh/storage/provenance/tasks.py 77

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

The archive: a (giant) Merkle DAG



Outline

The archive is ready and growing

Our current sources

- GitHub
- Debian, GNU
- WIP: Gitorious, Google Code, Bitbucket

The archive is ready and growing

Our current sources

- GitHub
- Debian, GNU
- WIP: Gitorious, Google Code, Bitbucket

The biggest source code archive already, … and growing daily!

Web API

First public version of our Web API (Feb 2017)
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) deposit of source code bundles directly to 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) deposit of source code bundles directly to 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!

Outline

Sharing the Software Heritage vision

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
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

You can help!

Coding

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

Working groups

- wiki.softwareheritage.org/index.php?title=Working_groups — working groups

Join us

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

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*

Questions?