Software Heritage and Guix
Software Heritage to the rescue of reproducible Science

Valentin Lorentz (@vlorentz) / Antoine R. Dumont (@ardumont)

Software Engineers, Software Heritage
Inria

16/09/2022
Event 10 years of Guix, Paris 2022
Outline

1. Introduction: the Software Heritage project
2. Reference archived code with SWHIDs
3. Collaboration Guix / SWH
4. Enters… Disarchive
5. Current Work in Progress
6. The End
What is SoftwareHeritage?

The Universal Source Code Archive

**Software Heritage**

**THE GREAT LIBRARY OF SOURCE CODE**
Why an archive? Software is spread all around

Many development platforms (popular forges: Guix, PyPI, npm, …)

Various distribution places (standalone forges: gitlab, heptapod, cgit, gitea…)

Projects tend to migrate from one place to another over time
Why an archive? Software is spread all around

- many development platforms (popular forges: Guix, PyPI, npm, …)
- various distribution places (standalone forges: gitlab, heptapod, cgit, gitea…)
- projects tend to migrate from one place to another over time

Fashion victims

One place…

… where can we find, track, search and recover all source code?
Why an archive? Software is fragile

Like all digital information, FOSS is fragile

- link rot: projects are created, moved around, removed
- data rot: physical media with legacy software decay
- business-driven code loss (e.g. Gitorious, Google Code, Bitbucket, …)
Why an archive? Software is fragile

Like all digital information, FOSS is fragile

- link rot: projects are created, moved around, removed
- data rot: physical media with legacy software decay
- business-driven code loss (e.g. Gitorious, Google Code, Bitbucket, …)

If a website disappears you go to the Internet Archive…

where do you go if (a repository on) GitHub or GitLab goes away?
Main Objectives

- Collect, Preserve and Share
Reference catalog

find and reference all software source code
Collect / Preserve

Reference catalog

**find and reference all software source code**

Universal archive

**preserve forever archived software source code**
Research infrastructure

- enable analysis of software source code
- make every piece identifiable
- and freely available...
Research infrastructure

- enable analysis of software source code
- make every piece identifiable
- and freely available...

Reproducibility

- ... exactly as it was when archived (as much as possible)
- for all research software artefacts
Global development history permanently archived in a uniform data model:
- over 12 billion unique source files from over 180 million software projects
- ~900 TB (uncompressed) blobs, ~25 B nodes, ~300 B edges
1. Introduction: the Software Heritage project
2. Reference archived code with SWHIDs
3. Collaboration Guix / SWH
4. Enters… Disarchive
5. Current Work in Progress
6. The End
Contents

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc.  <https://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, present or future, and that you know
what is happening to your programs as they run.

We wish to make sure that a free program remains free for
all its users. We define freedom in this context to mean that

1. the program’s users have the freedom to run, study how the
   program works, and change it,

2. the program’s distributors must give the users these freedoms.

To protect the users’ freedom, we need to prevent programs from
being altered, and to prevent others from stopping people from
altering it. For this reason, the license, called `copyleft’, requires
that copyrighted software remain free.

We have chosen to apply this license only when it is feasible to do so.
In some cases, this has not been feasible, as in the case of proprietary
networks, for example.

The exact terms and conditions for propagating free software, as
well as the consequences, are described in the GNU General Public
License. That license accompanies this manual.

sha1: 8624bcdae55baeef...
sha256: 8ceb4b9ee5aded...
sha1_git: 94a9ed024d385...
length: 35147
Data Model: A worked example
Directories

- .gitignore
- AUTHORS
- LICENSE
- MANIFEST.in
- Makefile
- Makefile.local
- README.db_testing
- README.dev
- bin
- deb
- docs
- requirements.txt
- setup.py
- sql
- swh
- utils

id: 515f00d44e92c65322aa9bf3fa097c00dd9c7d
Data Model: A worked example
Revisions

<table>
<thead>
<tr>
<th>Details</th>
<th>Changes</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA: 53334dc6b56e37f3ee426ba091092c267f9f6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Author: Nicolas Dandrimont <a href="mailto:nicolas@dandrimont.eu">nicolas@dandrimont.eu</a> (Thu Sep 1 14:26:13 2018)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Committer: Nicolas Dandrimont <a href="mailto:nicolas@dandrimont.eu">nicolas@dandrimont.eu</a> (Thu Sep 1 14:26:13 2018)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subject: provenance.tasks: add the revision -&gt; origin cache task</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parent: fc3a8b59ca1df424d860f29ab07ffee4dc35d10 : test...storage: properly pipeline origin and cont...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>provenance.tasks: add the revision -&gt; origin cache task</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

swh/storage/provenance/tasks.py

---

```bash
tree 515f00d44e92c65322aaa9bf3fa097c00dbb9c7d
parent fc3a8b59ca1df424d860f29ab07ffee4dc35d10
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
```
Data Model: A worked example
Data Model: A worked example

Valentin Lorentz (@vlorentz) / Antoine R. Dumont (@ardumont)

Snapshots

commit 08f1eeb25770109525eb3ce2163466c53a1d9158 refs/heads/alive
commit ba4432a24e3f9f6e323ae6c2921e6f4e4ebe61c57eb refs/heads/directory-listing-arrays
commit d69e0dbf89238fffe589a927aee1c8d5d7273389d3c5 refs/heads/too
commit c7ff9eaa8e8b227f804690f5a08019f6d7e45b8e88 refs/heads/master
commit 7ca197fc6d2d82e404584b1ed49e8bba4361a07f6e2 refs/heads/tmp-directory-addr
commit 6d42a285f3f7d0e5850ba5d427b53ee4fa2252e82e refs/heads/tmp/generic-releases
tag 29f043bc37c7f768a9665779f9f46907c75f7755 refs/tags/v0.0.1
>tag 72a19f9a13b0ae3e9499dd0b6b87f5b89ee72aa2ee2cd refs/tags/v0.0.10
>tag 3909c0ebabbb7fe8e5b37f65f233fbfa4fa5c0c refs/tags/v0.0.11
>tag 33378427a493ba56a967778bd8e5664fbc555 refs/tags/v0.0.12
>tag 06f7465275b327c7f59o3312cbfa9a35cf3b4035d refs/tags/v0.0.13
>tag 5a63252fe86ab545bc7a74466799a1e372f3bd refs/tags/v0.0.14
>tag 586feb4e588b47f5a0b593967643c3bca3ac77f refs/tags/v0.0.15
>tag 8c06885f4198bf5e3e17772bd28a2f66e5b5e51c refs/tags/v0.0.16
>tag a042a443e31f9be35e7257f3c503889a8b7c3d6 refs/tags/v0.0.17
>tag 228a2f16550d1d12222e556559462e1e6fc4993d9 refs/tags/v0.0.18
>tag 69637949ca4544977cf62d4a0d9de82655e4f7c refs/tags/v0.0.19
>tag 3b2f5a35f0c2a532b0a665103a5dd3823c275a67 refs/tags/v0.0.20
>tag 3147c3d1e46c6492f8f81e900b2127edbeff2c7 refs/tags/v0.0.21
>tag 215e5a58aba111e08e6b072e76b46b873a8790f refs/tags/v0.0.21
>tag 3b26b07c2d52e6221205a5edecf85f3f32d refs/tags/v0.0.22
>tag 8cbeee8d4a473c5d42789e460a16ac3c72aba4 refs/tags/v0.0.23

id: b464cad1b66fff266a37b46ea6e7a04b545e904b
Meet the SWHID intrinsic identifiers

- **swh:**
- **cnt:**
- **41ddb23118f92d7218099a5e7a990cf58f1d07fa**

```
- **origin_ctx:**
  - `:origin=https://github.com/chrislgarry/Apollo-11`

- **visit_ctx:**
  - `:visit=swh:1:snp:206c27c0c031c6aac6b5feddb8fe082dea9836`

- **anchor_ctx:**
  - `:anchor=swh:1:rev:3913f198f4383d4d638c0485d6aa902ff2f35828`

- **path_ctx:**
  - `:path=/Luminary099/BURN_BABY_BURN--MASTER_IGNITION_ROUTINE.agc`

- **lines_ctx:**
  - `:lines=64-72`
```
Outline

1. Introduction: the Software Heritage project
2. Reference archived code with SWHIDs
3. Collaboration Guix / SWH
4. Enters… Disarchive
5. Current Work in Progress
6. The End
How does this relate to Guix?

- Nothing is eternal, source code (in all forms) disappears
- Hopefully, SWH keeps a copy of everything
- **Guix ensures source code is archived in SWH when building** ("Save Code Now")
- After source code actually disappears, falls back to SWH when rebuilding ("Software Heritage Vault")
2018: Guix / SWH to ensure source code artifacts are pushed to SWH
2019: TWEAG / Guix / SWH: Work on a new loader to regularly ingest new artifacts
2022: ongoing work to refactor current loader into a standard lister/loader
Reproducibility is of the essence!

Current situation

- Persistent intrinsic identifiers (SWHID) are not (yet?) package manager standard
- Guix (and other) package managers reference tarball hashes, instead of hashes of the content

```scheme
(define-public ... 
  (package ... 
    (source (origin (method url-fetch) 
      (uri (string-append "https://..." version ".tar.gz"))) 
      (sha256 (base32 "03mwi1l3354x52nar...")))) 
  ...
```
Reproducibility is of the essence!

**Current situation**

- Persistent intrinsic identifiers (SWHID) are not (yet?) package manager standard
- Guix (and other) package managers reference tarball hashes, instead of hashes of the content

```lisp
(define-public ...
  (package ...
    ...
      (source (origin (method url-fetch)
                      (uri (string-append "https://..." version ".tar.gz")))
             (sha256 (base32 "03mwi1l3354x52nar...")))...)```

**Solutions**

- make (non-specific swh) SWHID standard or rebuild original bit-for-bit tarball
How to rebuild original tarballs?

**pristine-tar**
- [https://manpages.debian.org/bullseye/pristine-tar/pristine-tar.1.en.html](https://manpages.debian.org/bullseye/pristine-tar/pristine-tar.1.en.html)
- `xdelta`: binary diffs of tar headers’ content and order
- `zgz`: guessing compression parameters

**Limitations**
- it is brittle (relies on reference Tar implementation)
- it produces large diffs
How it started

Discussions

- "gforge.inria.fr to be taken off-line in Dec. 2020"
  https://issues.guix.gnu.org/42162
- "lookup ingested tarballs by container checksum"
  https://forge.softwareheritage.org/T2430

New software

- Disarchive by Timothy Sample https://git.ngyro.com/disarchive/
How it works:

Principles

- Manifest of tarball fields: entry order, PAX headers, ...
- References the root directory in SWH
- WIP: guessing compression parameters/implementations (using zgz)
- -> rebuild original .tar, then original .tar.{gz,xz}
(disarchive
 (version 0)
 (tarball
  (name "test-archive.tar")
  (digest (sha256 "0da9fa3e7b360533678338871d9dd36f3...")))
 (default-header
  (chksum (trailer " "))
  (magic "ustar ")
  (version " \x00")
  (devmajor 0 (source "" (trailer "")))
  (devminor 0 (source "" (trailer "")))
  (data-padding ""))

...
Example manifest (2/2)

(disarchive
...

(headers
  ("test-archive/" (mode 493) (chksum 4291) (typeflag 53))
  ("test-archive/file-a" (size 15) (chksum 4849))
  ("test-archive/file-b" (size 15) (chksum 4850)))
(padding 6656)
(input (directory-ref
  (version 0)
  (name "test-archive")
  (addresses
    (swhid "swh:1:dir:902b1e94f0f5efdde6..."))
  (digest (sha256 "277decb2666f4832ef64a..."))))
Planned integration of SWH with Disarchive

Currently
- SWH does not store Disarchive manifests yet

Plan
- Run Disarchive every time SWH loads a tarball
- Add the manifest to the Archive
- when someone requests `tarball-hash`, rebuild from the manifest
NixGuix manifests coverage in SWH

**goal: 100% coverage**

- currently missing sources due to technical limitations: bare files, directories, patches
- Redesign in progress to deal with such limitations
Integration

- code dump at https://git.ngyro.com/swh/
- needs to be reviewed and merged
Outline

1. Introduction: the Software Heritage project
2. Reference archived code with SWHIDs
3. Collaboration Guix / SWH
4. Enters... Disarchive
5. Current Work in Progress
6. The End
The End

Links
- Software Heritage website
- SWH’s Archive
- Development forge
- guix lint

We are hiring
We are hiring devs and sysadmins

Questions?
And thanks for your time!