Software Heritage for Open Science and Open Source

a revolutionary infrastructure

Roberto Di Cosmo
Réunion DGDS

Director, Software Heritage
Inria and Université de Paris Cité

May 25th 2023
So/f_tware: a pillar of Open Science

Software powers modern research

Over 20% of articles across all disciplines share software

2023 French Open Science Monitor

Key pillar: software

Nota Bene

software may be a tool, a research outcome and a research object

access to the source code is essential!

Preserving (the history of) source code is necessary for reproducibility
How are we managing our software?

Reproducibility, maintenance in Academia

Security, integrity, traceability in Industry

Can they track the software that they

- ship, use, acquire
- has that bug or vulnerability

Awareness is raising at the level of public policy

(articles: here, here, here and here)
International highlights

Paris Call on Software Source code (2019, UNESCO)

40 international experts call to "promote software development as a valuable research activity, and research software as a key enabler for Open Science/Open Research, [...] recognising in the careers of academics their contributions to high quality software development, in all their forms"

Open Source in UNESCO recommendations for Open Science, 2021

Software in the EOSC

2020 EOSC SIRS connect scholarly ecosystem via Software Heritage
2021 EOSC Task Force on Infrastructures for Research Software
2022 FAIRCORE4EOSC project WP6 implements SIRS report
2023 INFRAEOSC call on quality of scientific software

And much more

Software track in OSEC 2022, Software working group launched in Science Europe, DFG adds software to model CV (9/22), NASA unveils Open Science policy (12/22), ...
2nd National Plan for Open Science (6/7/2021)

**Open and promote research software source code**

- **actions (selection)**
  - charter for research software policy
  - recognize software development (see 2022 prize and 2023 call)
  - coordinate communities of practice
  - connected ecosystem of research outputs

- **recommendations (selection)**
  - archive in Software Heritage
  - standardise and use SWHID
  - build a national catalog of research software
  - leverage ADAC network

See official announcement
1. Context
2. Supporting the software pillar of Open Science
3. Software Heritage for Open Science
4. Software Heritage for (research on) Open Source
5. Meet the Software Heritage dataset(s)
6. Efficient traversal of the full graph
7. Impact on ESE studies
8. Perspectives and news
9. Conclusion
A plurality of needs

Researchers
- archive and reference software used in articles
- find useful software
- get credit for developed software
- verify, reproduce, improve results

Laboratories/teams
- track software contributions
- produce reports
- maintain web page

Research Organizations and/or Funders
- know its software assets
- technology transfer
- impact metrics
- funding strategy
- career evaluation
Basic needs

**Archive**
Research software artifacts must be properly **archived**
make sure we can **retrieve** them (**reproducibility**)

**Reference**
Research software artifacts must be properly **referenced**
make sure we can **identify** them (**reproducibility**)

**Describe**
Research software artifacts must be properly **described**
make it easy to **discover** and **reuse** them (**visibility**)

**Cite/Credit**
Research software artifacts must be properly **cited** (**not the same as referenced!**)
to give **credit** to authors (**evaluation!**)
Some popular approaches, and why they do not fit the bill

A - Since the 1970’s 1990’s
.zip or .tar file on:
- ftp-server (e.g. gnu)
- web page (example)
- document archive (+ DOI sample)

B - Since the 2000’s
Rely on software forges
- institutional/project (e.g. example)
- free commercial ones: BitBucket, GitHub, GitLab, … (e.g. parmap)

C: a mix of the two
Can get no satisfaction…

A Poor user experience
B No preservation guarantee
C Can do so much better
Forges are *not* archives!

2015: the first big bad news

Google Code and Gitorious.org shutdown: ~1M endangered repositories
- broken links in the web of knowledge (my papers too)

Big bad news keep coming in

- summer 2019: BitBucket announces Mercurial VCS sunset
- july 2020: BitBucket erases 250,000+ repositories (including research software)
- summer 2022: GitLab.com considers erasing all projects that are inactive for a year

In Academia too!

- 2021: Inria’s old gforge is unplugged… breaks the Opam build chain for OCaml

We need a universal archive of software source code: now we have one!
1. Context
2. Supporting the software pillar of Open Science
3. Software Heritage for Open Science
4. Software Heritage for (research on) Open Source
5. Meet the Software Heritage dataset(s)
6. Efficient traversal of the full graph
7. Impact on ESE studies
8. Perspectives and news
9. Conclusion
Collect, preserve and share all software source code

Preserving our heritage, enabling better software and better science for all

Reference catalog
find and reference all software source code

Universal archive
preserve and share all software source code

Research infrastructure
enable analysis of all software source code
A peek under the hood: a universal archive

Global development history permanently archived in a uniform data model

- over 14 billion unique source files from over 210 million software projects
- ~1PB (compressed) blobs, ~30 B nodes, ~400 B edges

R. Di Cosmo  roberto@dicosmo.org  (CC-BY 4.0)

Software Heritage, Open Science and Open Source  May 25th 2023  10 / 29
A walkthrough

- **Browse and Reference** (e.g. Apollo 11, and your work may be already there!)
- **Trigger archival**, use the `updateswh` browser extension, configure the webhooks
- **Cite with** biblatex-software ([CTAN, Overleaf ACMART template](https://mirror.ctan.org/info/lshort/laTeX/acmart/acmart.pdf))
- **Describe with** Codemeta (use codemeta generator)
- **Curated deposit in SWH via HAL**, see for example: LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, …
- **Extracting all the software products** for Inria, for CNRS, for CNES, for LIRMM or for Rémi Gribonval using HalTools
- **Example with Parmap**: devel on Github, archive in SWH, curated deposit in HAL
- **Example research articles**:
  - compare Fig. 1 and conclusions in the 2012 version and the updated version
  - SWHID in a replication experiment
Adoption indicators

From Melissa Harrison’s OSEC 2022 talk

What are they “referencing”?

<table>
<thead>
<tr>
<th>source</th>
<th>n</th>
<th>percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not available</td>
<td>2868</td>
<td>46.22</td>
</tr>
<tr>
<td>GitHub</td>
<td>1151</td>
<td>18.55</td>
</tr>
<tr>
<td>software heritage</td>
<td>387</td>
<td>6.24</td>
</tr>
<tr>
<td>zenodo</td>
<td>142</td>
<td>2.29</td>
</tr>
<tr>
<td>r package</td>
<td>70</td>
<td>1.13</td>
</tr>
<tr>
<td>cran</td>
<td>56</td>
<td>0.90</td>
</tr>
<tr>
<td>r package version</td>
<td>54</td>
<td>0.87</td>
</tr>
<tr>
<td>github</td>
<td>35</td>
<td>0.56</td>
</tr>
</tbody>
</table>

- 6205 “software” references identified
- Top 8 listed, then long tail of 1055 other sites – 932 are unique “source”

HAL+SWH in the Open Science software booklet

Use on replicabilitystamp.org

b/Surf: Interactive Bézier Splines on Surface Meshes

Claudio Mancinelli, Giacomo Nazzaro, Fabio Pellacini, Enrico Puppo
IEEE Transactions on Visualization and Computer Graphics (TVCG)

Funding agencies recommendations ANR 2023 guidelines (p. 17)

Enfin, conformément au 2ème Plan national pour la science ouverte, L’ANR recommande que les logiciels développés durant le projet soient mis à disposition sous une licence libre et que les codes sources soient stockés dans l’archive Software Heritage en indiquant la référence au financement ANR.
Archiving and referencing

For all source code used in research (yes, even small scripts!)
- archive and reference in Software Heritage (see detailed HOWTO)

Describing and Citing/Crediting

For software one wants to put forward, add these extra steps:
- add codemeta.json with description (see the codemeta generator)
- (french partners) reference in the HAL portal (see online HAL documentation)
- cite software using the biblatex-software package (in CTAN and TeXLive)

We can (and must)
- train students and colleagues
- engage journals, conferences, learned societies
Outline

1. Context
2. Supporting the software pillar of Open Science
3. Software Heritage for Open Science
4. Software Heritage for (research on) Open Source
5. Meet the Software Heritage dataset(s)
6. Efficient traversal of the full graph
7. Impact on ESE studies
8. Perspectives and news
9. Conclusion
Where does reused software come from?

Do you know where it comes from?

- the software you ship
- the software you use
- the software you acquire
- the software that
  - has that bug
  - has that vulnerability

KYSW: Know Your SoftWare

Like KYC in banking, KYSW is now essential all over IT…

Sec. 4. Enhancing Software Supply Chain Security

ensuring and attesting, to the extent practicable, to the integrity and provenance of open source software

May 2021 POTUS Executive Order
1. Context
2. Supporting the software pillar of Open Science
3. Software Heritage for Open Science
4. Software Heritage for (research on) Open Source
5. Meet the Software Heritage dataset(s)
6. Efficient traversal of the full graph
7. Impact on ESE studies
8. Perspectives and news
9. Conclusion
The full graph in the AWS Open Data collection

https://registry.opendata.aws/software-heritage/

Software Heritage Graph Dataset

Description
Software Heritage is the largest existing public archive of software source code and accompanying development history. The Software Heritage Graph Dataset is a fully deduplicated Merkle DAG representation of the Software Heritage archive. The dataset links together file content identifiers, source code directories, Version Control System (VCS) commits tracking evolution over time, up to the full states of VCS repositories as observed by Software Heritage during periodic crawls. The dataset's contents come from major development forges (including GitHub and GitLab), FOSS distributions (e.g., Debian), and language-specific package managers (e.g., PyPI). Crawling information is also included, providing timestamps about when and where all archived source code artifacts have been observed in the wild.

Update Frequency
Data is updated yearly

License
Creative Commons Attribution 4.0 International. By accessing the dataset, you agree with the Software Heritage Ethical Charter for using the archive data and the terms of use for bulk access.

Documentation
https://docs.softwareheritage.org/devel/swh-dataset/graph/athena.html

Managed By
Software Heritage

Resources on AWS

Description
Software Heritage Graph Dataset

Resource type
S3 Bucket

Amazon Resource Name (ARN)
arn:aws:s3:::softwareheritage

AWS Region
us-east-1

AWS CLI Access (No AWS account required)
aws s3 ls --no-sign-request s3://softwareheritage/

Description
S3 Inventory files

Resource type
S3 Bucket

Amazon Resource Name (ARN)
arn:aws:s3:::softwareheritage-inventory

AWS Region
us-east-1

AWS CLI Access (No AWS account required)
aws s3 ls --no-sign-request s3://softwareheritage-inventory
A peek at the dataset

Accessing graph leaves (a.k.a. contents)

$ aws s3 ls --no-sign-request s3://softwareheritage/
  PRE content/
  PRE graph/

File contents can be accessed using their SHA1 checksum

$ aws s3 cp --no-sign-request \
  s3://softwareheritage/content/8624bcdae55baeef00cd11d5dfcfa60f68710a02 .

Notice that file contents are compressed:

$ zcat 8624bcdae55baeef00cd11d5dfcfa60f68710a02 | head

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.
A peek at the dataset, cont’d

Annual dumps of (inner nodes of) the full graph

$ aws s3 ls --no-sign-request s3://softwareheritage/graph/

2018-09-25/
2019-01-28-popular-3k-python/
2019-01-28-popular-4k/
2020-05-20/
2020-12-15/
2021-03-23-cpython-3-5/
2021-03-23-popular-3k-python/
2021-03-23/
2022-04-25/

How to use

- online full documentation
- Antoine Pietri’s PhD Thesis

How to cite

Antoine Pietri, Diomidis Spinellis, Stefano Zacchiroli. The Software Heritage Graph Dataset: Public software development under one roof. MSR 2019. (bibtex)
Example: most popular commit verbs (stemmed)

Query using Amazon Athena

```sql
SELECT COUNT(*) AS C, word FROM (SELECT word_stem(lower(split_part(trim(from_utf8(message)),' ','1'))) AS word FROM revision WHERE length(message) < 1000000) WHERE word != '' GROUP BY word ORDER BY C DESC LIMIT 20;
```

Total cost: approximately .5 euros

Results (20)

<table>
<thead>
<tr>
<th>#</th>
<th>C</th>
<th>word</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>271573294</td>
<td>updat</td>
</tr>
<tr>
<td>2</td>
<td>163328012</td>
<td>merg</td>
</tr>
<tr>
<td>3</td>
<td>140044381</td>
<td>add</td>
</tr>
<tr>
<td>4</td>
<td>105800317</td>
<td>fix</td>
</tr>
<tr>
<td>5</td>
<td>103646653</td>
<td>ad</td>
</tr>
<tr>
<td>6</td>
<td>52891401</td>
<td>bump</td>
</tr>
<tr>
<td>7</td>
<td>50067041</td>
<td>initi</td>
</tr>
<tr>
<td>8</td>
<td>45609622</td>
<td>creat</td>
</tr>
<tr>
<td>9</td>
<td>42633225</td>
<td>remov</td>
</tr>
<tr>
<td>10</td>
<td>32230842</td>
<td>chang</td>
</tr>
<tr>
<td>11</td>
<td>23110410</td>
<td>delet</td>
</tr>
<tr>
<td>12</td>
<td>20734745</td>
<td>new</td>
</tr>
<tr>
<td>13</td>
<td>18644508</td>
<td>comnt</td>
</tr>
<tr>
<td>14</td>
<td>15651821</td>
<td>test</td>
</tr>
</tbody>
</table>
Outline

1. Context
2. Supporting the software pillar of Open Science
3. Software Heritage for Open Science
4. Software Heritage for (research on) Open Source
5. Meet the Software Heritage dataset(s)
6. Efficient traversal of the full graph
7. Impact on ESE studies
8. Perspectives and news
9. Conclusion
State-of-the-art graph compression from social networks

Paolo Boldi, Antoine Pietri, Sebastiano Vigna, Stefano Zacchiroli
Ultra-Large-Scale Repository Analysis via Graph Compression
SANER 2020, 27th Intl. Conf. on Software Analysis, Evolution and Reengineering. IEEE

Results

Full graph structure (25 B nodes, 350 B edges) in 200 GiB RAM
- traversal time is tens of ns per edge
- bidirectional traversals implemented
- beware: metadata access is still off RAM

Java and gRPC APIs available

docs.softwareheritage.org/devel/swh-graph/grpc-api.html
Examples

Find all origins containing a given content

```plaintext
```

Gives a list of origins including "https://github.com/rdicosmo/parmap", encoded as "swh:1:ori:8903a90c8f07159be7aed69f19d66d33db3f86" (beware: this is not a SWHID!)

Shortest provenance path of a content in a given origin

```plaintext
grpc_cli call localhost:50091 swh.graph.TraversalService.FindPathBetween "\nsrc: 'swh:1:ori:8903a90c8f07159be7aed69f19d66d33db3f86', \ndst: 'swh:1:cnt:8722d84d658e5e11519b807abb5c05bfbf531f0', \nmask: {paths: ['swhid']}" | egrep 'swhid'
```

connecting to localhost:50091

```
swhid: "swh:1:ori:8903a90c8f07159be7aed69f19d66d33db3f86"
swhid: "swh:1:snp:1527a93b039d70f6a78b05d76b77c6209912887"
swhid: "swh:1:rev:82df563aefc86b9164e7d10d40f2d8c4b1c78d"
swhid: "swh:1:dir:484db39bb2825886191837bb0960b7450f909bb"
swhid: "swh:1:dir:4d15e44b378e39dd23817abbe756cd47ad14575"
swhid: "swh:1:cnt:8722d84d658e5e11519b807abb5c05bfbf531f0"
```

Rpc succeeded with OK status
Mining Android Applications on Software Heritage

RQ: how to build a specific dataset for a given research question?

- Specific and limited API
- Hardly reproducible
- Generic, single, fine-grained and unlimited API
- Growing number of source codes
- Easy to update the dataset

(from the Inria/IRISA DiverSE team)
Using the SWH merkle dag to identify android repositories

Identify android application repositories = Find the AndroidManifest.xml among the sources

1) Iterate over the graph nodes until you find a directory node containing a file named "AndroidManifest.xml".
Using the SWH merkle dag to identify android repositories

Identify android application repositories = Find the AndroidManifest.xml among the sources

2) Extract the SWH identifier of the blob corresponding to the AndroidManifest.xml and download the corresponding file through the SWH Web API
Using the SWH merkle dag to identify android repositories

Identify android application repositories = Find the AndroidManifest.xml among the sources

3) Traverse the graph in backward direction to the origin node and get the repository url
**Bottomline**

- Broad variety of sources in *one open dataset* reduces usual GH bias
- Reference simple *standard data format* VCS and forge details are abstracted away
- Simplifies reproducibility packages no need to create a full copy, *just list the SWHIDs!*
- Software Heritage does the heavy lifting for you no need to scrape/download repositories all over again
1. Context

2. Supporting the software pillar of Open Science

3. Software Heritage for Open Science

4. Software Heritage for (research on) Open Source

5. Meet the Software Heritage dataset(s)

6. Efficient traversal of the full graph

7. Impact on ESE studies

8. Perspectives and news

9. Conclusion
## Research projects

### Open science (EOSC framework)
- **FAIR-IMPACT**: recommendations for research software
- **FAIRCORE4EOSC**
  - connect SWH with Zenodo (InvenioRDM), Dataverse, Dagstuhl, episcience, OpenAire, swMath, …)
  - SWH mirror for the EOSC

### Cybersecurity
- **SWHSec (PTCC)**: IMT, CEA, SU, Inria approved, starting now infra for research

### Big Code
- CINECA, ENEA, Unibo, UniPi around Leonardo and the Bologna mirror submitted

... and much more!

please come onboard
A revolutionary infrastructure

The graph of public software development

All software development in a single graph ...

- enable traceability

The global ledger of public code

... a Merkle graph

- ensure integrity

A pillar of Open Science

Reference archive of Research Software

- reproducibility
- reference

Reference platform for Big Code

uniform data structure

- large scale studies
- machine learning, AI, ...

R. Di Cosmo roberto@dicosmo.org (CC-BY 4.0)
A rally flag for a grand vision

Bring together academia, industry, governments, communities

"to build a reference, global infrastructure for open and better software"

Software Heritage is the first brick …

- vendor neutral
- open source
- a worldwide initiative
- a long term initiative

… that will enable

- archival, reference, integrity
- qualification, sharing and reuse
- a global software knowledge base
- test and deploy world class tooling

A lot more is needed

Software Heritage can be the catalyser of a way bigger undertaking

You can help!

help maintain and improve the infrastructure, adapt research tools to work with it, …
The floor is yours

Let’s work together!

Questions?

References

- UNESCO, *Draft recommendations on Open Science* 2021, (online)