What would you do with billions of source code files?

News from the Software Heritage project

Roberto Di Cosmo
roberto@dicosmo.org

October 5th 2017
LIG
Outline

1. Software is everywhere and nowhere
2. Source code is essential...
3. … but it is not in good shape!
4. The Software Heritage initiative
5. Building the network
6. Technical status
7. A glimpse of the archive
8. Selected research challenges: building the archive
9. Selected research challenges: using the archive
10. Conclusion
Software is everywhere

At the heart of our society

- communication, entertainment
- administration, finance
- health, energy, transportation
- education, research, politics
- ...

Knowledge enabler

- Key mediator for accessing all information
- Essential component of modern scientific research

Software embodies our collective Knowledge and Cultural Heritage
Outline

1. Software is everywhere and nowhere
2. Source code is essential…
3. … but it is not in good shape!
4. The Software Heritage initiative
5. Building the network
6. Technical status
7. A glimpse of the archive
8. Selected research challenges: building the archive
9. Selected research challenges: using the archive
10. Conclusion
"The source code for a work means the preferred form of the work for making modifications to it."

— GPL Licence

Hello World

Program (source code)

/* Hello World program */

#include<stdio.h>

void main()
{
    printf("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
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)

```c
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalves = 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 * ( threehalves - ( x2 * y + y ) ); // 1st iteration
    // y = y * ( threehalves - ( x2 * y + y ) ); // 2nd iteration, this can be removed
    return y;
}
```

Net. queue in Linux (excerpt)

```c
/*
 * SFB uses two B[I][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 "Bln" */
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.”
1. Software is everywhere and nowhere
2. Source code is essential...
3. ... but it is not in good shape!
4. The Software Heritage initiative
5. Building the network
6. Technical status
7. A glimpse of the archive
8. Selected research challenges: building the archive
9. Selected research challenges: using the archive
10. Conclusion
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?
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

If you study the stars, you go to Atacama...

... where is the very large telescope of source code?
So/f_tware is everywhere and nowhere

Source code is essential...

... but it is not in good shape!

The Software Heritage initiative

Building the network

Technical status

A glimpse of the archive

Selected research challenges: building the archive

Selected research challenges: using the archive

Conclusion
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

- Reference repository
- Provenance
- Certification
- Security

- Reproducibility
- Traceability
- Open Access
- Software studies

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

Cultural Heritage
Industry
Research
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
The Knowledge Conservancy Magic Triangle

Legenda (links are important!)

- articles: ArXiv, HAL, …
- data: Zenodo, …
- software: Software Heritage to the rescue
1. Software is everywhere and nowhere
2. Source code is essential…
3. … but it is not in good shape!
4. The Software Heritage initiative
5. Building the network
6. Technical status
7. A glimpse of the archive
8. Selected research challenges: building the archive
9. Selected research challenges: using the archive
10. Conclusion
Our principles

Open approach
open source, transparency

In for the long haul
non profit, replication

Thomas Jefferson, February 18, 1791

…let us save what remains: not by vaults and locks which fence them from the public eye and use in consigning them to the waste of time, but by such a multiplication of copies, as shall place them beyond the reach of accident.
Sharing the Software Heritage vision

See more

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

April 3rd, 2017: landmark Inria Unesco agreement…

https://www.softwareheritage.org/blog

September 28, 2017 International Day for Universal Access to Information
Mauritius Call on software heritage
Outline

1. Software is everywhere and nowhere
2. Source code is essential…
3. … but it is not in good shape!
4. The Software Heritage initiative
5. Building the network
6. Technical status
7. A glimpse of the archive
8. Selected research challenges: building the archive
9. Selected research challenges: using the archive
10. Conclusion
Archive coverage

Our current sources

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

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

The *richest* source code archive already, … and growing daily!
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 (for now)

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

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

Software Heritage Archive

Merkle DAG + blob storage

Distro

Package repos

Forges

GitHub lister

Git lister

Git

Mercurial loader

Debian package loader

Debian source

PyPi lister

tar loader

tar

svn

svn

hg

hg

git

git

git

dsc
dsc
tar
tar

Listing (full/incremental)

Loading & deduplication

Scheduling

Billions of source code files

Roberto Di Cosmo
Much more than an archive!

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 archive in a few pictures

A giant (extended) Merkle DAG
Outline

1. Software is everywhere and nowhere
2. Source code is essential...
3. ... but it is not in good shape!
4. The Software Heritage initiative
5. Building the network
6. Technical status
7. A glimpse of the archive
8. Selected research challenges: building the archive
9. Selected research challenges: using the archive
10. Conclusion
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/
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
  }, ...
]
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.../"
    }
  ]
}
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
1. Software is everywhere and nowhere
2. Source code is essential...
3. ... but it is not in good shape!
4. The Software Heritage initiative
5. Building the network
6. Technical status
7. A glimpse of the archive
8. Selected research challenges: building the archive
9. Selected research challenges: using the archive
10. Conclusion
Data compression

Deduplication is performed at the file level *across all projects in the world*.

**Pros**
- very efficient to cope with file clones
- quite resilient to technology changes

**Cons**
- a minor edit creates two different files

**Challenge: exploit file similarities**
- adapt / improve variable size checksums
- compression rates of up to 100 to 1 may arise
Metadata alignment

Many concepts related to source code

- project, archive, source, language, licence, bts, mailing list, …
- developer, committer, author, architect, …

Many existing ontologies

DOAP, FOAF, Appstream, schema.org, ADMS.SW, …

Many disparate catalogs

Freecode (40.000+), Plume (400+), Debian (25.000+), OpenHub (670.000+), …

Challenge: scale up metadata to millions of projects

- reconcile existing ontologies
- link and check existing catalogs with Software Heritage
- handle inconsistent data and provenance information
Software phylogenetics

The Software Diaspora

- Code often *migrates* across projects: forks, copy-paste
- Code gets *cloned*: reuse, language limitations, code smells
- Projects *migrate* across forges: fashion, functionality
- Projects get *cloned*: mirrors, packages

Challenge: tracing software evolution across billions of files

- rebuild the history of software artefacts
- identify code origins
- spot code clones
- build project impact graphs
Distributed infrastructure

The software graph

- files
- directories
- commits
- projects

all de-duplicated in Software Heritage

Challenge: design efficient architectures and algorithms

- replication and availability (CAP?)
- navigation
- query
- path analysis
1. Software is everywhere and nowhere
2. Source code is essential...
3. ... but it is not in good shape!
4. The Software Heritage initiative
5. Building the network
6. Technical status
7. A glimpse of the archive
8. Selected research challenges: building the archive
9. Selected research challenges: using the archive
10. Conclusion
Code search: an old problem

A natural need
- Find the definition of a function/class/procedure/type/structure
- Search examples of code usage in an archive of source code
- You name it…

A natural approach
- Regular expressions

We have all used grep since the 1970’s!

where is the challenge?
How do we search in millions of source code files?

Google code search (open 2006, closed 2011)

see https://swtch.com/~rsc/regexp/regexp4.html reborn in 2013 for Debian http://sources.debian.net/

how

- build an inverted index of trigrams from all source files
- map regexps to trigrams
- filter files that may match
- run grep on each file (using the cloud)

performance

scaled reasonably well up to 1 billion lines of codes
Challenge: scaling up code search

What about *all the source code* in the world?

Software Heritage is *two orders of magnitude* bigger already
- over *two billion* unique source files
- *hundreds* of billions of LOCs

We need new insight for handling this.

Beyond regular expressions?

Advanced code search requires
- language specific *patterns*
- working on *abstract syntax trees*

Regular expressions are a nice *swiss-army knife* approximation, can we build a specific tool that scales?
Software as Big Data

Remember the numbers

- 60+ million repositories ingested
- 700+ million commits
- 3+ billion unique source files / 200 TB of raw source code

and growing by the day!

Challenge: what can machines learn here?

- programming patterns / trends
- developer skills
- vulnerabilities
- bugs and fixes
### Remember the numbers

- 60+ million repositories ingested
- 700+ million commits
- 3+ billion unique source files / 200 TB of raw source code

and growing by the day!

### Challenge: can we make this fit in memory?

- efficient graph representation
- fast non-local queries
- mitigate the size/speed tradeoff
Software is everywhere and nowhere

Source code is essential...

... but it is not in good shape!

The Software Heritage initiative

Building the network

Technical status

A glimpse of the archive

Selected research challenges: building the archive

Selected research challenges: using the archive

Conclusion
Learn more

- a reference archive of all FOSS ever written
- a fantastic new tool for research software
- a unique complement for development platforms
- an international, open, nonprofit, mutualized infrastructure


Contributing

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

Questions?