

Software Heritage

Building the Universal Software Archive for Open Science

Roberto Di Cosmo

`roberto@dicosmo.org`

March 15th, 2018



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

- 1 Software is everywhere around us
- 2 Software source code for Science!
- 3 The Software Heritage initiative
- 4 Status
- 5 Building for the long term
- 6 Conclusion



At the heart of *our society*



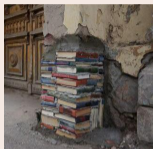
- communication, entertainment
- administration, finance
- health, energy, transportation
- education, research, politics
- ...

At the heart of *technology*

- house appliances \approx 10M SLOC
- phones \approx 20M SLOC, *cars* \approx 100M SLOC
- Internet of things, ...



Key mediator for accessing *all* information (c) Banski



Information is a main pillar of our modern societies.

Absent an ability to correctly interpret digital information, we are left with [...] "rotting bits" [...] of no value.

Vinton G. Cerf IEEE 2011

Software is *an essential component* of modern scientific research

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

Top 100 papers (Nature, October 2014)

Software embodies our *Knowledge and Cultural Heritage*

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");
}
```

Harold Abelson, Structure and Interpretation of Computer Programs (1st ed.)

1985

“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[1][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 also 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.”

~ 50 years, a lightning fast growth

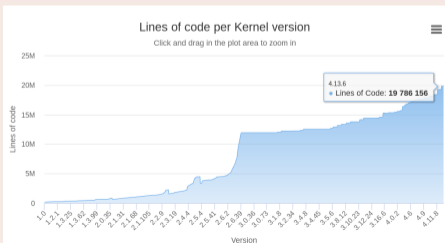
Apollo 11 Guidance Computer (~60.000 lines), 1969



"When I first got into it, nobody knew what it was that we were doing. It was like the Wild West."

Margaret Hamilton

Linux Kernel



... now in your pockets!

are we taking care of all this?

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?



A word cloud of terms related to software fragility and digital preservation. The most prominent words are 'damage', 'disaster', 'malicious', 'obsolete', 'attack', 'deletion', and 'format'. Other visible words include 'media', 'aging', 'tear', 'dependencies', 'dangling', 'wear', 'corruption', 'encryption', 'reference', and 'storage'. The words are arranged in a roughly circular pattern, with 'damage' at the top and 'format' at the bottom.

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?

- 1 Software is everywhere around us
- 2 Software source code for Science!
- 3 The Software Heritage initiative
- 4 Status
- 5 Building for the long term
- 6 Conclusion



We face a science crisis

"Sub-prime science"? (Nicholas Humphrey)

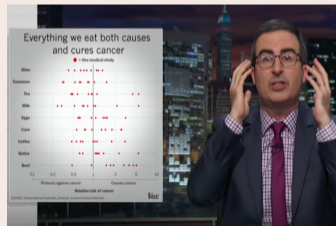


- inconsistencies
- data corruption, fraud
- non reproducible findings... (picture from Nature, Sep. 2015)

The world starts noticing



October 2013



John Oliver, *Science* May 2016

How we built our scientific knowledge

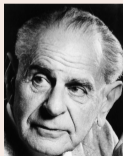
The experimental method



- 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

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*?

Analysis of 613 papers

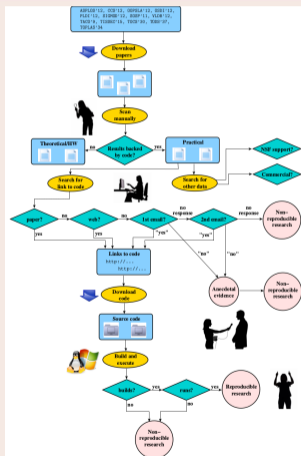
- 8 ACM conferences: ASPLOS'12, CCS'12, OOPSLA'12, OSDI'12, PLDI'12, SIGMOD'12, SOSP'11, VLDB'12
- 5 journals: TACO'9, TISSEC'15, TOCS'30, TODS'37, TOPLAS'34

all very practical oriented

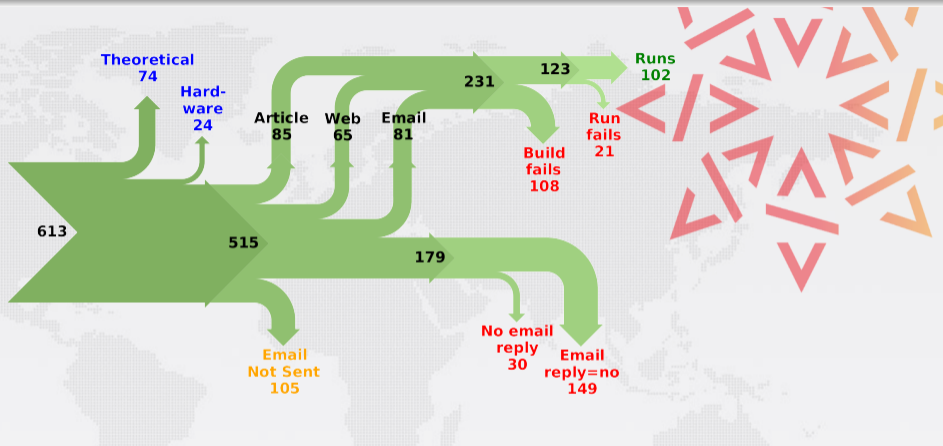
The basic question

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!

URL decay disrupts the *web of reference*

Web links *are not* permanent (even *permalinks*)

there is no general guarantee that a URL... which at one time points to a given object continues to do so

T. Berners-Lee et al. Uniform Resource Locators. RFC 1738.

404

URLs used in articles *decay!*

Analysis of *IEEE Computer* (Computer), and the *Communications of the ACM* (CACM): 1995-1999

- the *half-life* of a referenced URL *is approximately 4 years* from its publication date
D. Spinellis. The Decay and Failures of URL References.

Communications of the ACM, 46(1):71-77, January 2003.

Similar findings in Lawrence, S. et al. *Persistence of Web References in Scientific Research*, IEEE Computer, 34(2), pp. 26-31, 2001.

An example from Astronomy

Domain	links (broken)	.html	.txt	.dat	.gz	.tar	.fits	tilde
oac.harvard.edu	802 (110)	336 (70)	0	0	4 (2)	5 (4)	1	0
heasarc.gsfc.nasa.gov	640 (33)	423 (27)	1	0	0	0	0	0
www.stsci.edu	498 (61)	205 (29)	3	0	0	0	0	15 (10)
esc.harvard.edu	471 (152)	212 (99)	0	0	0	0	0	1 (1)
ssc.spitzer.caltech.edu	427 (194)	125 (76)	3 (3)	0	0	0	0	0
cfa-www.harvard.edu	352 (68)	277 (52)	1	0	0	0	0	54 (17)
archive.stsci.edu	308 (58)	57 (9)	2	1 (0)	0	0	0	0
www.ipac.caltech.edu	285 (14)	209 (12)	0	0	0	0	0	0
www.atnf.csiro.au	211 (21)	12 (6)	0	0	0	0	0	7 (5)
space.mit.edu	193 (10)	58 (5)	1	0	0	0	0	2 (1)
www.astro.psu.edu	186 (4)	103 (1)	1	10 (1)	1	1	0	2
www.eso.org	186 (58)	54 (22)	1 (1)	0	0	0	0	4 (1)
isa.ipac.caltech.edu	163 (5)	38	0	0	1	0	0	0
www.sdss.org	156 (2)	106 (1)	0	0	0	0	0	0
hea-www.harvard.edu	125 (37)	42 (17)	1	0	0	1	0	26 (16)
physics.nist.gov	125 (3)	63 (2)	0	0	0	0	0	0
www.noao.edu	120 (3)	50 (2)	0	0	0	0	0	0
vmm.vilspa.esa.es	118 (35)	23 (19)	0	0	8 (1)	0	0	1 (1)
www.astro.princeton.edu	115 (31)	43 (14)	0	0	0	0	0	53 (12)
ad.usno.navy.mil	110 (27)	98 (22)	3 (3)	0	0	0	0	1 (1)

This table lists total number of links and broken links (HTTP status codes 3xx, 4xx, and 5xx) to top domains (domains with over 100 links) found within articles published in the four main astronomy journals between 1997 and 2008. The table also shows, for each domain, the portion of links to common filename extensions, as well as links that contain the tilde character.

doi:10.1371/journal.pone.0104798.t001

How Do Astronomers Share Data?

Pepe, Goodman, Muench, Crosas, Erdmann

[dx.doi.org/10.1371/journal.pone.0104798](https://doi.org/10.1371/journal.pone.0104798)

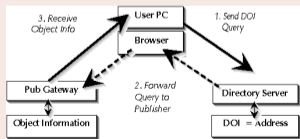
PLOS August 28, 2014

DOI limitations

Example: `doi:10.1109/MSR.2015.10`

- to find what `10.1109/MSR.2015.10` is, go to a *resolver* (e.g. `doi.org`)
- this returns `http://ieeexplore.ieee.org/document/7180064/`
- at this URL we find ...

Architecture of the DOI infrastructure



- DOI resolution *can change*
- content at URL *can change*
- no *intrinsic* way of noticing
- persistence based on *good will* of multiple parties

No catalog, no archive, no references, ... and we are at a turning point

Looking at the past

- a lot of old software misplaced, lost, or behind barriers, but...
- most founding fathers are still here, and willing to share
- **urgent** to collect their knowledge

Only a few years left.

Looking at the future

- software development and use skyrockets: more programmers, and more code!
- **essential** to provide a **universal** platform for all the future software source code

Every year that goes by makes the problem worse.

it is **urgent** to take action!

- 1 Software is everywhere around us
- 2 Software source code for Science!
- 3 The Software Heritage initiative**
- 4 Status
- 5 Building for the long term
- 6 Conclusion





Software Heritage



Our mission

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

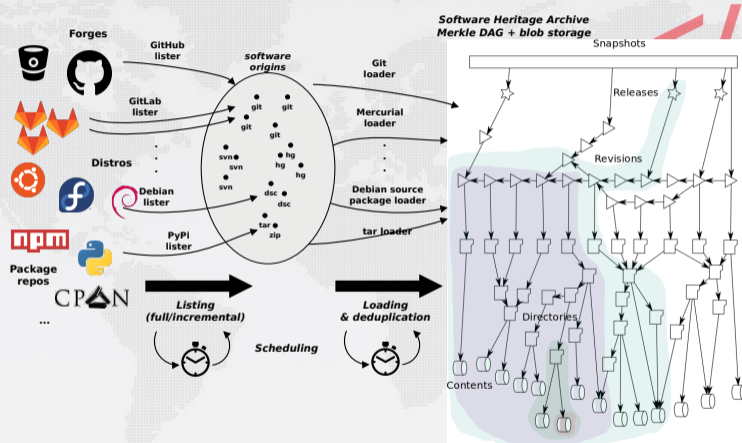
Past, present and future

Preserving the past, enhancing the present, preparing the future

- 1 Software is everywhere around us
- 2 Software source code for Science!
- 3 The Software Heritage initiative
- 4 **Status**
- 5 Building for the long term
- 6 Conclusion

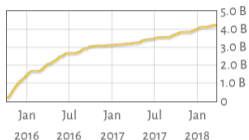


Architecture (simplified)



Source files

4,290,063,587



Commits

980,310,191



Projects

83,797,945

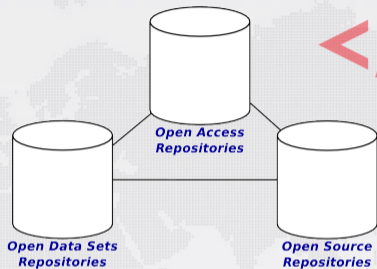


Current sources

- live: GitHub, Debian
- one-off: Gitorious, Google Code, GNU
- WIP: Bitbucket

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

The *richest* public source code archive, ... and growing daily!



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**

Browsing the archive contents

- archive.softwareheritage.org (fosdem/2018)

Archiving scientific software via HAL

- open on the Inria instance, see the deposit guide at <http://bit.ly/swhdeposithalen>

- 1 Software is everywhere around us
- 2 Software source code for Science!
- 3 The Software Heritage initiative
- 4 Status
- 5 Building for the long term
- 6 Conclusion



Cultural Heritage



Industry



Research



Education



Software Heritage

Open approach

- Transparency
- Free Software
- User and contributor community building

Objectiveness

- Facts and provenance
- *Intrinsic* identifiers
- Full development history

Long term

- Multi-stakeholder
- Nonprofit
- Replication *at all layers*

Three pillars

Science and technology

- build on sound basis
- fantastic playground for research

Resources

- fund the effort
- transfer to industry and society

Awareness

- promote public and private policies
- community building

Sponsoring Software Heritage work

Inria
INVENTEURS DU MONDE NUMÉRIQUE

>= 100Ke/year	 Microsoft	 intel	 SOCIÉTÉ GÉNÉRALE	
>= 50Ke/year				
>= 25Ke/year		 HUAWEI		
>= 10Ke/year	<small>Data Archiving and Networked Services</small>  DANS	 NOKIA Bell Labs	 ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA	 GitHub

Sharing the Software Heritage vision



See more

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

April 3rd, 2017: landmark Inria Unesco agreement...

Inria
INVENTEURS DU MONDE NUMÉRIQUE



<https://www.softwareheritage.org/blog>

September 28th, 2017

Mauritius Call on information access

- 1 Software is everywhere around us
- 2 Software source code for Science!
- 3 The Software Heritage initiative
- 4 Status
- 5 Building for the long term
- 6 Conclusion



Come in, we're open!



Software Heritage

www.softwareheritage.org

[@swheritage](https://twitter.com/swheritage)

Everybody is needed

become a mirror

our code

research

funding

<mailto:roberto@dicosmo.org>

forge.softwareheritage.org

use cases, new challenges

sponsorship.softwareheritage.org

Library of Alexandria of code



Take *urgent* action to

- recover the past
 - founding fathers still here
- structure the future
 - programming skyrockets

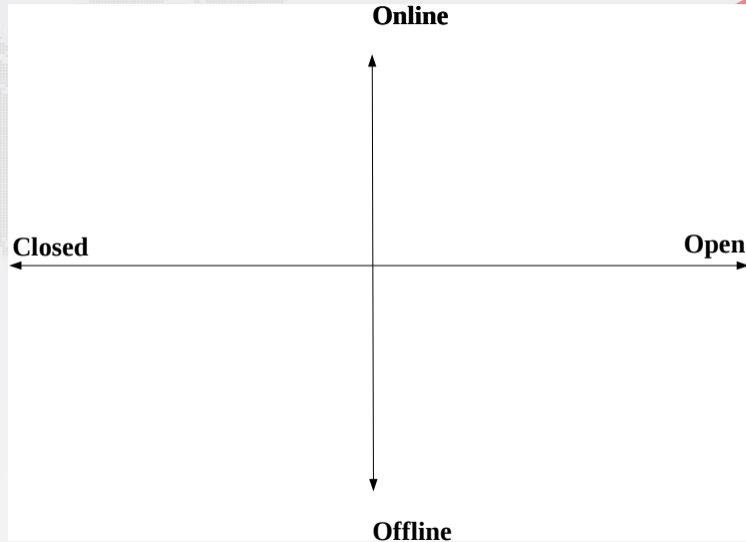
A CERN for Software



Photo: ALMA(ESO/NAOJ/NRAO), R. Hills

Build a *common infrastructure*

- supporting industry needs
- enabling software research
- fostering better science
- for society as a whole



Online, open source code: automation overview

