What would you do with billions of source code files?
Challenges and opportunities in software archival

Roberto Di Cosmo
roberto@dicosmo.org

21 Juin 2016
Journées Scientifiques Inria

Software Heritage
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*
Software is spread all around

- many disparate development platforms
- a myriad places where distribution may happen
- projects tend to migrate from one place to the other over time
- URLs decay, DOIs are fragile

One place to bind them...

... where can we find, track and search all the source code?
Software is missing its own Research Infrastructure

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

A wealth of software research on crucial issues...

- safety, security; test, verification, proof;
- software engineering, software evolution;
- empirical and big data studies;

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

... where is the very large telescope of source code?
The Software Heritage Project

Software Heritage
PRESERVING TECHNICAL KNOWLEDGE

Our mission

Collect, organise, preserve and share the source code of all the software that lies at the heart of our culture and our society.

Past, present and future

Preserving the past, enhancing the present, preparing the future.
“Programs must be written for people to read, and only incidentally for machines to execute.” Harold Abelson, *Structure and Interpretation of Computer Programs*

**Distinguishing features**

- *executable* and *human readable* knowledge (an *all time new*):
  - even hardware is... software! (VHDL, FPGA, …)
  - *text files are forever*
- naturally *evolves* over time
  - the *development history* is key to its *understanding*
- complex: large *web of dependencies*, millions of SLOCs

**In a word**

- software *is not just another* sequence of bits
- a software archive *is not just another* digital archive
The Knowledge Conservancy Magic Triangle

Legenda (links are important!)
- articles: ArXiv, HAL, …
- data: Zenodo, …
- software: Software Heritage to the rescue
## The people

### Core team
- Roberto Di Cosmo
- Stefano Zacchioli
- Nicolas Dandrimont
- Antoine Dumont

### Scientific advisors
- Serge Abiteboul
- Jean-François Abramatic
- Gerard Berry
Where we are today: technically

Our sources
- GitHub — all public repositories, as of April 2016
- Debian — daily snapshots of all suites since 2005–2015
- GNU — all historical releases up to August 2015
- Gitorious — retrieved full mirror from Archive Team
- Google Code — retrieved full mirror from Google

Some numbers
- 21 million repositories ingested (10M next in line)
- 500 million commits
- 2.5 billion unique source files / 200 TB of raw source code

Here are some research challenges arising from all this
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+), FramaSoft (1500+), 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
- synthesise missing information (machine learning)
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
- navigation
- what happens to CAP? (updates are nondestructive!)
- query
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?
Finding a needle in a haystack: size matters!

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?
So/f_tware as Big Data

Remember the numbers

- 21 million repositories ingested (10M next in line)
- 500 million commits
- 2.5 billion unique source files / 200 TB of raw source code

and growing by the day!

Challenge: what can machines learn here?

- programming patterns
- developer skills
- vulnerabilities
- bugs and fixes
Come in, we’re open

Software Heritage working groups

*Expanding, Interconnecting, Evolving, and Using* the archive

- go see [https://wiki.softwareheritage.org](https://wiki.softwareheritage.org)

Resources for distributed storage

- share storage/compute nodes for research use

Adoption

- help connecting Software Heritage with everyday’s work
- spread its use across research communities

Research

- take over some of the scientific challenges
Conclusion

Software Heritage is

- a revolutionary *reference archive* of all software ever written
- a unique *complement* for *development platforms*
- an international, open, nonprofit, *mutualized infrastructure*

*We need your help to make it happen*

Questions?

Keeping in contact

*

`mailing list: swh-science@inria.fr`

*https://sympa.inria.fr/sympa/info/swh-science*