

Reproducibility in Computer Science

Where we come from, where we are, where we go

Roberto Di Cosmo

Director, Software Heritage
Inria and Université Paris Cité

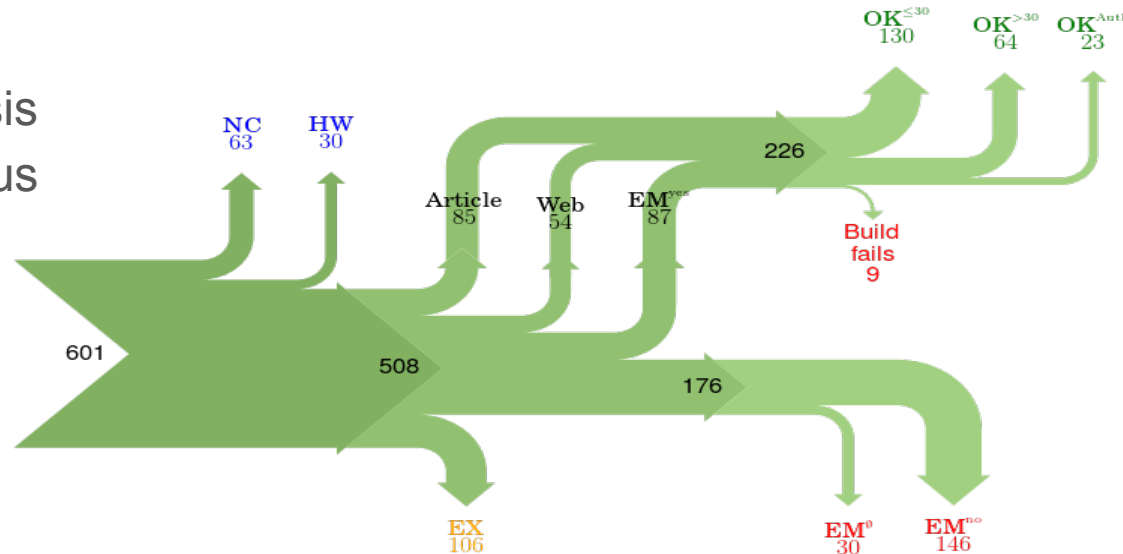
State of the art in the field ~2010

Software Engineering

2009: Carlo Ghezzi, 60% of ACM TOSEM papers have code, only 20% installable

Computer systems research

2014: Christian Collberg, analysis of **~600 papers** in prestigious venues, **~200 cannot even find the source code!**



Awareness and actions

Artifact Evaluation Committees

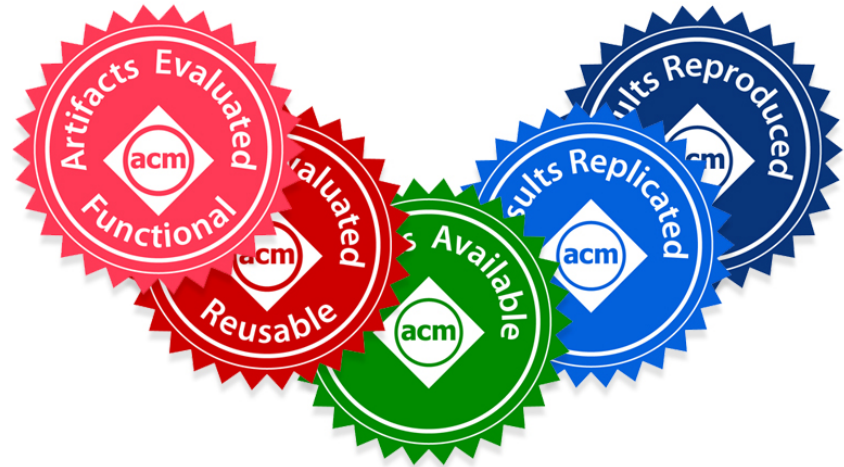
2011: run the first time as an award at ESEC-FSE ([J. Vouillon and R. Di Cosmo](#))

2012-today: [the process generalizes](#) to a [list too long to maintain](#)

ACM software badges for publications

See [home page](#) for details.

- Very good intentions, but ...
- Perfectible implementation



A few key issues in reproducibility (*there are many more!*)

Archive

Ensure **long term availability** of artifacts **with the development history**

Reference

Ensure **precise identification** of artifacts at **various levels of granularity**

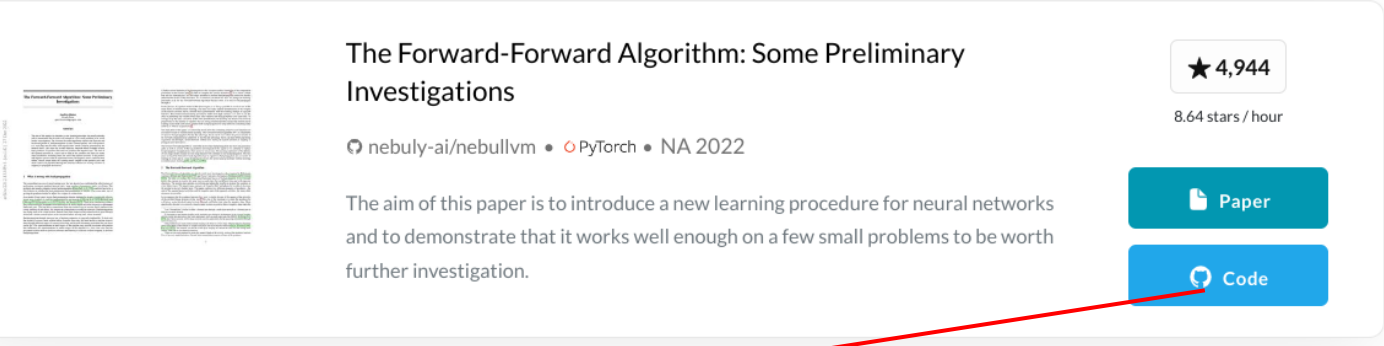
Describe

Provide **detailed description** (machine readable metadata)

and **proper documentation** (build instructions, dependencies, configuration)

and also *link to relevant papers*

Not there yet, event for these basic needs: Papers with code



The Forward-Forward Algorithm: Some Preliminary Investigations

nebulu-ai/nebullvm • PyTorch • NA 2022

The aim of this paper is to introduce a new learning procedure for neural networks and to demonstrate that it works well enough on a few small problems to be worth further investigation.

★ 4,944
8.64 stars / hour

Paper

Code

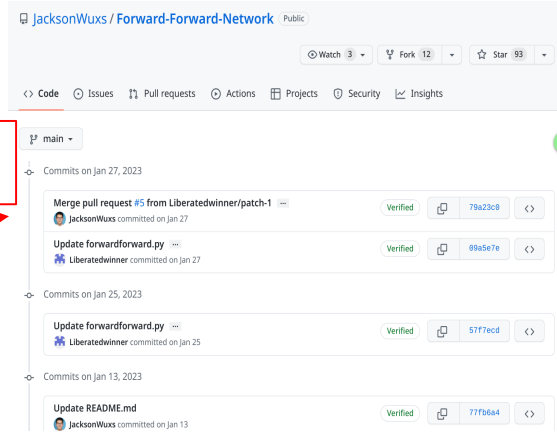
Not an archive!

Code

nebulu-ai/nebullvm	★ 4,944	PyTorch
keras-team/keras-io	★ 2,128	TensorFlow
mohammadpz/pytorch_forward_forward	★ 1,190	PyTorch
JacksonWuxs/Forward-Forward-Network	★ 93	PyTorch
EscVM/EscVM_YT	★ 48	PyTorch

See all 6 implementations

Which version?



JacksonWuxs / Forward-Forward-Network Public

Code Issues Pull requests Actions Projects Security Insights

main

Commits on Jan 27, 2023

- Merge pull request #5 from Liberatedwinner/patch-1
Verified 79a23c8
- Update forwardforward.py
Verified 09a5e7e

Commits on Jan 25, 2023

- Update forwardforward.py
Verified 5f77ecd

Commits on Jan 13, 2023

- Update README.md
Verified 77fb6a4

We can and must do better: archive in Software Heritage

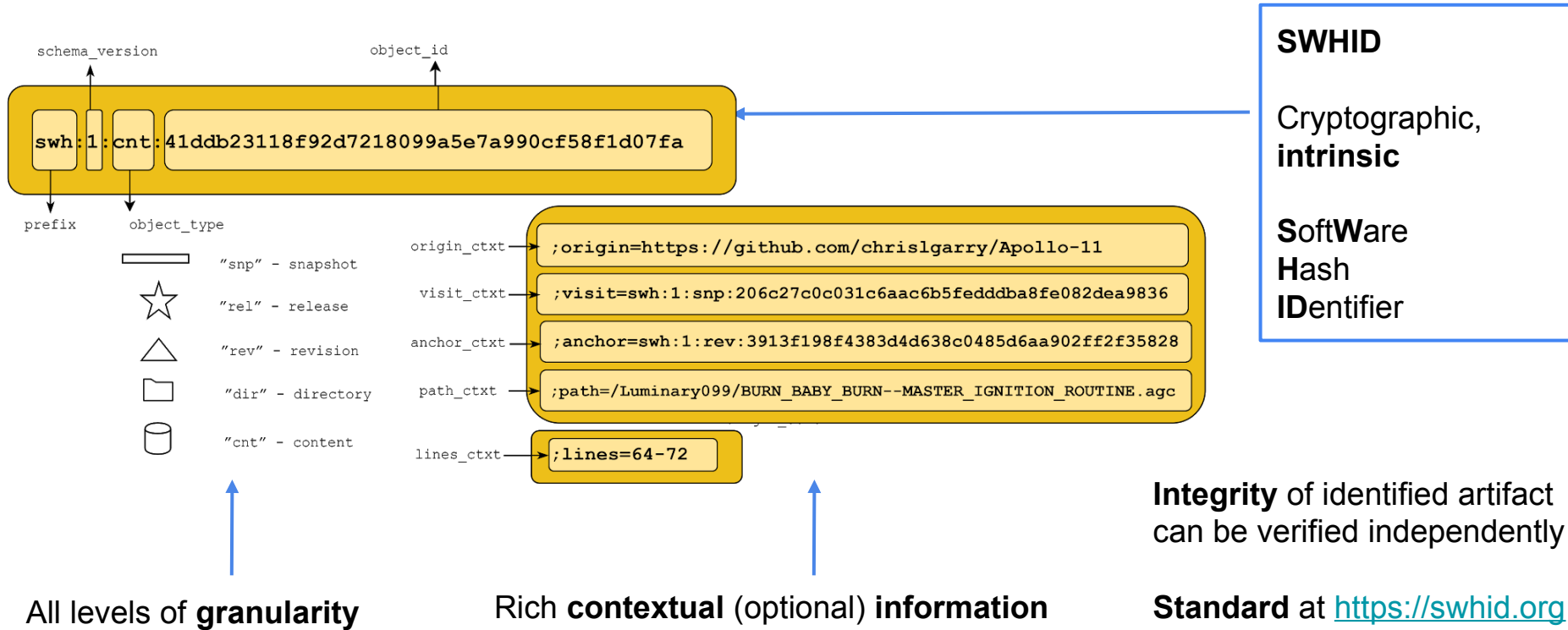
The screenshot shows the GitHub repository page for 'rdicosmo / parmapp'. The repository is public and has 89 stars and 20 forks. The main content area shows a commit by Roberto Di Cosmo on Nov 25, 2022, with 288 likes. The commit message is 'Update biblatex snippet'. The commit details show a file named 'Update biblatex snippet' with a diff view. The repository description states: 'Parmapp is a minimalistic library allowing to exploit multiverse architect'. The repository has 89 stars, 5 watchers, and 20 forks. There are 12 releases, with the latest being 'Update for OCaml 5.0' on Jan 2.

The screenshot shows the Software Heritage archive page for the repository. The page title is 'Browse the archive'. The URL is 'https://github.com/rdicosmo/parmapp'. The page shows the commit history, with the selected commit being '28 February 2023, 01:55:26 UTC'. The commit message is 'Update biblatex snippet'. The page shows the file structure of the repository, including 'config', 'example', 'src', 'tests', '.gitignore', 'AUTHORS', and 'LICENSE'. The file sizes are listed: '.gitignore' (38 bytes) and 'AUTHORS' (722 bytes).

- Regular crawling
- **One click** archival via **updateswh** browser extension
- Webhooks for BitBucket, Gitea, GitHub, GitLab, Sourceforge

The screenshot shows a tweet by Gabriel Altay (@gabrielaltay). The tweet text is: 'Just realized @Bitbucket disabled all mercurial repositories when the @asclnet informed me that a link associated with an old paper of mine was down. Thought all was lost, but someone archived all the repos! very classy move by @octobus_net and @SWHeritage.' The tweet is dated 1:48 AM · 31 août 2020 · Twitter Web App.

We can and must do better: **reference** in Software Heritage



We can and must do better: **reference** in Software Heritage

Getting the SWHID for a code fragment

You can also get the SWHID of a file, or a code fragment inside a file. For this, navigate first to the file, select (optionally) the code fragment of interest by clicking on the line number of the first line, and shift-clicking on the line number of the last line. Then, pull out the red Permalinks tab and copy the SWHID identifier or the corresponding permalink.

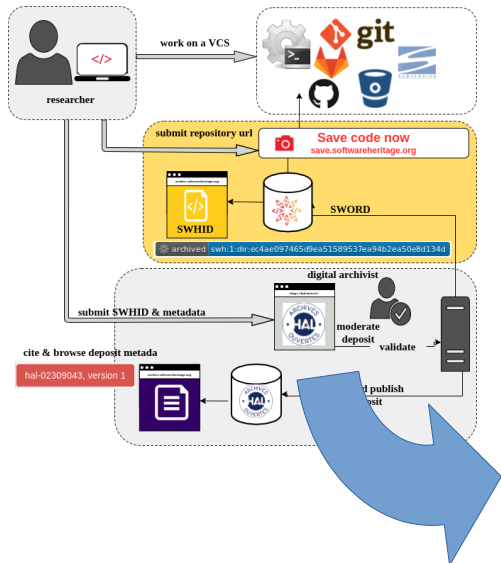
~ 30 billion SWHIDs can be found in Software Heritage

```
66
67 let can_redirect path =
68   if not(Sys.file_exists path) then
69     try
70       Unix.mkdir path 0o777; true
71     with Unix.Unix_error(e,_,s) ->
72       (* another job may have created it between the time we called file_exists
73        * and here *)
74     if e == Unix.EEXIST then true
75     else begin
76       (Printf.eprintf "[Pid %d]: Error creating directory '%s'
77        * without stdout/stderr\n" (Unix.getpid ()) path (Unix.error_message e))
78       false
79     end
80   else true
81
82 let log_debug fmt =
83   Printf.kprintf (
84     if !debug_enabled then begin
85       (fun s -> Format.eprintf "[Parmap]: %s@." s)
86     else ignore
87   ) fmt
88
89 (* freopen emulation, from Xavier's suggestion on OCaml
90  *
91  * let reopen_out outchan path fname =
92  *   if can_redirect path then
93  *     begin
94  *       flush outchan;
95  *       let filename = Filename.concat path fname in
96  *       let fd1 = Unix.descr_of_out_channel outchan in
97  *       let fd2 = Unix.openfile
```

All levels of **granularity**:

- repository snapshot
- release
- revision
- directory
- file content
- code fragment

In France : HAL + Software Heritage for describe and cite



<https://hal.archives-ouvertes.fr/ha>

Free and accessible knowledge

Home | Submit | Browse- | Search | Documentation

hal-02130801, version 1

LinBox

The LinBox Group 1 2 3 4 5 6 7 8 9 Details

- 1 ECO - Exact Computing
- 2 ARIC - Arithmetic and Computing
- 3 AVALON - Algorithms and Software Architectures for Distributed and HPC Platforms
- 4 CIS - Department of Computer and Information Sciences [Newark]
- 5 Drexel University
- 6 NCSU - Department of Mathematics [Raleigh]
- 7 United States Naval Academy
- 8 SCG - Symbolic Computation Group
- 9 CAS-C - Calcul Algébrique et Symbolique, Sécurité, Systèmes Complexes, Codes et Cryptologie

LJK - Laboratoire Jean Kuntzmann

Abstract: LinBox is a C++ template library of routines for solution of linear algebra problems including linear system solution, rank, determinant, minimal polynomial, characteristic polynomial, and Smith normal form. Algorithms are provided for matrices with integer entries or entries in a finite field. A number of matrix storage types is provided, especially for blackbox representation of sparse or structured matrix classes. A few algorithms for rational matrices are available. LinBox also uses underlying data structures and algorithms for integer, rational, polynomial, finite fields and rings, as well as dense and sparse matrix formats coming from the Givaro (<https://caays.gricad-pages.univ-grenoble-alpes.fr/givaro>) and FFLAS-FFPACK (<http://linbox-team.github.io/flas-ffpack>) libraries.

Document type: Software

Domain: Computer Science [cs] | Symbolic Computation [cs.SC]

Complete list of metadata | Display

BROWSE

Software Heritage swh1:dir:393b611a1424f032e83569b6762502371cfcf65.origin=https://hal.archives-ouvertes.fr/ha-02130801/visit=swh:1:snp:19c29b988fe02623c707f4c0b991f42481e691fb.anchor=swh:1:rev:e818328952266b7875c692963b11963b1496107/path=1

Browse

Browse the archive Enter a SWHID to resolve or keyword(s) to search for it

<https://hal.archives-ouvertes.fr/ha-02130801>

14 June 2019, 13:43 UTC

<> Code Branches (1) Releases (0) Visits

Revision: e818328952266b7875c692963b11963b1496107 393b611 / linbox-1.6.3 / linbox / config-blas.h Raw File

Tip revision: e818328952266b7875c692963b11963b1496107 authored by Software Heritage on 11 June 2019, 08:12 UTC
hal: Deposit 297 in collection hal

config-blas.h

```
1 /* config-blas.h
2  * Copyright (C) 2005 Pascal Giorgi
3  *          2007 Clement Pernet
4  * Written by Pascal Giorgi <pgiorgi@waterloo.ca>
5  *
6  * =====LICENCE=====
7  * This file is part of the library LinBox.
8  *
9  * LinBox is free software: you can redistribute it and/or modify
10 * it under the terms of the GNU Lesser General Public
11 * License as published by the Free Software Foundation; either
12 * version 2.1 of the License, or (at your option) any later version.
13 *
14 * This library is distributed in the hope that it will be useful,
15 * but WITHOUT ANY WARRANTY; without even the implied warranty of
16 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
17 * Lesser General Public License for more details.
18 *
19 * You should have received a copy of the GNU Lesser General Public
20 * License along with this library; if not, write to the Free Software
21 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
22 * =====LICENCE=====
23
24
25
26 #ifndef LINBOX_config_blas_h
```

[swh:1:dir:393b611a1424f032e83569b6762502371cfcf65](https://hal.archives-ouvertes.fr/ha-02130801/visit=swh:1:dir:393b611a1424f032e83569b6762502371cfcf65)

The way ahead

Archival and reference for source code

- **Technical barriers** are mostly solved issues (*over 6 years of work*)
- **Social barriers** still stand in the way (adoption, training, cost mutualization, ...)

Thank you

- Software Heritage: <https://softwareheritage.org> and [the 2022 annual report](#)
- HOWTO archive, reference, describe and cite research software: <https://bit.ly/swh-howto-research>
- Software deposit and metadata curation: [HAL-SWH Webinar, July 2022](#)
- Deuxième plan national pour la Science Ouverte: [official website](#)
- Software Pillar session in OSEC 2022: [official website](#)
- EOSC SIRS report: <https://data.europa.eu/doi/10.2777/28598>
- Roberto Di Cosmo and Marco Danelutto. [Rp] Reproducing and replicating the OCamlP3I experiment. ReScience C, 6(1):#2, April 2020. [link]

Learn more