Preserving the global software heritage

lwn.net/Articles/693471/

By **Nathan Willis**
July 7, 2016

The Software Heritage initiative is an ambitious new effort to amass an organized, searchable index of all of the software source code available in the world (ultimately, including code released under free-software licenses as well as code that was not). Software Heritage was launched on June 30 with a team of just four employees but with the support of several corporate sponsors. So far, the Software Heritage software archive has imported 2.7 billion files from GitHub, the Debian package archive, and the GNU FTP archives, but that is only the beginning.

In addition to the information on the Software Heritage site, Nicolas Dandrimont gave a presentation about the project on July 4 at DebConf; video [WebM] is available. In the talk, Dandrimont noted that software is not merely pervasive in the modern world, but it has cultural value as well: it captures human knowledge. Consequently, it is as important to catalog and preserve as are books and other media—arguably more so, because electronic files and repositories are prone to corruption and sudden disappearance.

Thus, the goal of Software Heritage is to ingest all of software source code available, index it in a meaningful way, and provide front-ends for the public to access it. At the beginning, that access will take the form of searching, but Dandrimont said the project hopes to empower research, education, and cultural analysis in the long term. There are also immediate practical uses for a global software archive: the tracking of security vulnerabilities, assisting in license compliance, and helping developers discover relevant prior art.

The project was initiated by Inria, the French Institute for Research in Computer Science and Automation (which has a long history of supporting free-software development) and as of launch time has picked up Microsoft and Data Archiving and Networked Services (DANS) as additional sponsors. Dandrimont said that the intent is to grow Software Heritage into a standalone non-profit organization. For now, however, there is a small team of full-time employees working on the project, with the assistance of several interns.

The project's servers are currently hosted at Inria, utilizing about a dozen virtual machines and a 300TB storage array. At the moment, there are backups at a separate facility, but there is not yet a mirror network. The archive itself is online, though it is currently accessible only in limited form. Users can search for specific files by their SHA-1 hashes, but cannot browse.

Indices

It does not take much contemplation to realize that Software Heritage's stated goal of indexing all available software is both massive in raw numbers and complicated by the vast assortment of software sources involved. Software Heritage's chief technology officer (CTO) is Stefano Zacchiroli, a former Debian Project Leader who has recently devoted his attention to Debsources, a searchable online database of every revision of every package in the Debian archive.

Software Heritage is an extension of the Debsources concept (which, no doubt, had some influence in making the Debian archive one of the initial bulk imports). In addition to the Debian archive, at launch time the Software Heritage archive also included every package available through the GNU project's FTP site and an import of all public, non-fork repositories on GitHub. Dandrimont mentioned in his talk that the Software Heritage team is currently working with Google to import the Google Code archive and with Archive Team to import its Gitorious.org archive.

Between the three existing sources, the GitHub data set is the largest, accounting for 22 million repositories and 2.6 billion files. For comparison, in 2015, Debsources was reported to include 11.7 million files in just over 40,000 packages. Google Code included around 12 million projects and Gitorious around 2 million.

But those collections account for just a handful of sites where software can be found. Moving forward, Software Heritage wants to import the archives for the other public code-hosting services (like SourceForge), every Linux distribution, language-specific sites like the Python Package Index, corporate and personal software repositories, and (ultimately) everywhere else.

Complicating the task is that this broad scope, by its very nature, will pull in a lot of software that is not open-source or free software. In fact, as Zacchiroli confirmed in an email, the licensing factor is already a hurdle, since so many repositories have no licensing information:

There is a lot of publicly available source code (e.g., on GitHub) which is simply not licensed as FOSS, often due to the lack of a license. That stuff will become FOSS one day though, either when its license gets clarified (even retroactively), or when copyright expires.

The way I like to think about this is: we want to protect the entire Software Commons. Free/Open Source Software is the largest and best curated part of it; so we want to protect of FOSS. Given the long-term nature of Software Heritage, we simply go for all publicly available source code (which includes all of FOSS but is larger), as it will become part of the Software Commons one day too.

For now, Zacchiroli said, the Software Heritage team is focused on finalizing the database of the current software and on putting a reliable update mechanism in place. GitHub, for example, is working with the team to enable ongoing updates of the already imported repositories, as well as adding new repositories as they are created. The team is also writing import tools for use ingesting files from a variety of version-control systems (old and new).

Access

Although the Software Heritage archive's full-blown web interface has yet to be launched, Dandrimont's talk provided some details on how it will work, as well as how the underlying stack is designed.

All of the imported archives are stored as flat files in a standard filesystem, including all of the revisions of each file. A PostgreSQL database tracks each file by its SHA-1 hash, with directory-level manifests of which files are in which directory. Furthermore, each release of each package is stored in the database as a directed acyclic graph of hashes, and metadata is tracked on the origin (e.g., GitHub or GNU) of each package and various other semantic properties (such as license and authorship). At present, he said, the archive consists of 2.7 billion files occupying 120TB, with the metadata database taking up another 3.1TB. "It is probably the biggest distributed version-control graph in existence," he added.

Browsing through the web interface and full-text searching are the next features on the roadmap. Following that, downloading comes next, including an interface to grab projects with `git clone`. Further out, the project's plans are less specific, in part because it hopes to attract input from researchers and users to help determine what features are of interest.

At the moment, he said, the storage layer is fairly basic in its design. He noted that the raw number of files "broke Git's storage model" and that the small file sizes (3kB on average) posed its own set of challenges. He then invited storage experts to get involved in the project, particularly as the team starts exploring database replication and mirroring. The code used by the project itself is free software, available at forge.softwareheritage.org.

Because the archive contains so many names and email addresses, Zacchiroli said that steps were being taken to make it difficult for spammers to harvest addresses in bulk, while still making it possible for well-behaved users to access files in their correct form. "There is a tension here," he explained. The web interface will likely obfuscate addresses and the archive API may rate-limit requests.

The project clearly has a long road ahead of it; in addition to the large project-hosting sites and FTP archives, collecting all of the world's publicly available software entails connecting to thousands if not millions of small sites

and individual releases. But what Software Heritage is setting out to do seems to offer more value than a plain "file storage" archive like those offered by Archive Team and the Internet Archive. Providing a platform for learning, searching, and researching software has the potential to attract more investments of time and financial resources, two quantities that Software Heritage is sure to need in the years ahead.