Richtlinie zum „Software-Sprint“

ReproBuilds – Reproducible Builds in der Wirklichkeit

Schlussbericht

Zuwendungsempfänger:

Holger Levsen

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S31 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Seit Ende 2014 beschäftige ich mich mit Reproducible Builds, wobei wir uns bis zum Projektbeginn im wesentlichen damit beschäftigt haben, Reproducible Builds theoretisch möglich zu machen, d.h. wir haben Fehler(klassen) in Quellcode(s) analysiert und Patches bereitgestellt, und dies (verschiedenen intensiv) in verschiedenen (Linux und anderen) Distributionen. Ein Schwerpunkt meiner Arbeit innerhalb des Reproducible Builds Projektes liegt bei der Entwicklung und Betrieb des Services https://tests.reproducible-builds.org welches folgende Projekte kontinuierlich testet: Debian, Archlinux, OpenWrt, coreboot, NetBSD, FreeBSD und F-Droid.

Die bisherigen Tests und der Betrieb von https://tests.reproducible-builds.org richteten sich an die Entwickler von den Projekten, nicht aber an Userinnen und User.

Ziel des Protoypefund Projekt war es, die Entwicklung dahin zu lenken, Userinnen und User zu ermöglichen, selber zu entscheiden, ob sie nicht-reproduzierbare Software installieren.

Anders gesagt, Ziel war es, die theoretische Ebene für eines oder mehrere der getesteten Projekte zu verlassen und praktisch zu werden.

Beitrag des Projektes zu den Zielen der Förderinitative „Software-Sprint“

Freie Software ist die Basis des Software Sprints, da Freie Software reviewed, verbessert, geteilt und genutzt werden kann. (Die 4 Freiheiten von Freier Software.) Allerdings gibt es – außer Reproducible Builds – keine Möglichkeit, sicher zu sein, das die verwendete Software wirklich Freie Software ist, weil sich die 4 Freiheiten auf Sourcecode beziehen, während Binärcode verwendet wird und es – außer durch Reproducible Builds - nicht sichergestellt werden kann, das ein bestimmter Binärcode von einem bestimmen Sourcecode abstammt.

Ausführliche Darstellung der Ergebnisse

Mein erstes Ziel war die Erstellung von https://tests.reproducible-builds.org/archlinux/ wo die (theoretischen) Testergebnisse von Reproduzierbaren Archlinux untersucht und dargestellt werden sollten, um darauf aufbauend dann Archlinux praktisch reproduzierbar zu machen. Ersteres ist gelungen, zweiteres leider noch nicht, aufgrund der Probleme die ich weiter unten in „kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben“ (für Debian) beschrieben habe. Diese Probleme sind dieselben für Archlinux und Debian.

Ich betrachte diesen Teilschritt trotzdem als (teilweise) erfolgreich, weil er gezeigt hat/zeigt, daß

- a.) Archlinux sich auch reproduzierbar machen läßt (es also keine prinzipiellen Probleme gibt)
- b.) dazu geführt hat, diverse Probleme in Archlinux zu beheben und es
- c.) dadurch eine Einbindung der Archlinux Entwickler Community in das Reproducible Builds Projekt gegeben hat. Vor diesen Tests war ein Archlinux Entwickler Teil des Reproducible Projekts, heute sind 4-5 Leute aktiv.

Anfang Dezember 2018 war ich dann damit beschäftigt, den Reproducible Builds Summit in Paris auszurichten (siehe https://reproducible-builds.org/events/paris2018/ inkl. Einer ausführlichen Dokumentation der Ergebnisse des Summits). Ein technisch herausausragendes Ergebnisses dieses Summit war die Integration der (woanders erstellten) Ergebnisse der Tests von OpenSUSE in unsere SQL Projektdatenbank, mit dem Ziel diese Ergebnisse im JSON Format bereitzustellen, um dann Usertools zu entwickeln. Derzeit bieten wir JSON für Debian, Archlinux und OpenSUSE an (und seit dem grade vergangenen Wochenende auch für Alpinelinux). Die Entwicklung besagter Usertools befindet sich noch im Alphastudium aufgrund der weiter unten beschriebenen Probleme.

Anfang 2019 habe ich dann begonnen zu untersuchen, wie sich die 93% theoretische Reproduzierbarkeit von Debian in der Realität wiederspiegelt und bin zu der etwas deprimierenden Zahl von 54% real reproduzierbaren Paketen gekommen, was im wesentlichen an diesen drei Bugs liegt:

- #894441: binNMUs should be replaced by easy "no-change-except-debian/changelog-uploads" (as of 20190302, 12% of all binaries in Buster were binNMUs.)

- #900837: Mass-rebuild of packages for reproducible builds (blocked by #894441)
- #862538: security.debian.org: Please POST .buildinfo files to buildinfo.debian.net

Diese drei Bugs sind allerdings Probleme in der Debian Infrastruktur und Prozessen, die zum Einem eine Kooperation verschiedener Teams in Debian nötig machen und zum Anderen nicht während des Release-Freezes addressiert werden können. Debian produziert ca. alle zwei Jahre ein neues Release, dazu wird vorher die Distribution „eingefroren“, so daß nur minimal invasive Änderungen gemacht werden. Dieses Mal begann der Freeze am 20. Januar 2019, so daß wir diese Änderungen erst jetzt im Sommer 2019 nach dem Release vorantreiben können.

Die Erkenntnisse daraus führten nach Projektende dazu, daß ich https://buildinfos.debian.net aufgesetzt habe, weil der bestehende (und von einem anderem Teammitglied betriebene) https://buildinfo.debian.net Service für unsere Zwecke nicht ausreicht. Eine Zusammenführung dieser zwei ähnlichen Services steht noch aus, ist aber geplant.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Das Protoypefund Projekt hat meine Sicht auf das Reproducible Builds Projekt massiv verändert: dadurch das ich schon 5 Jahre an diesem Problem gearbeitet habe, bin ich quasi ein wenig betriebsblind geworden: theoretisch sind von 28000 Source Paketen zB in Debian „nur“ 93% reproduzierbar, d.h. wir haben noch ca. 1940 Source Pakete, die wir nicht reproduzieren können.

Diese 1940 Source Pakete sind ein (großes) Problem, bzw. 1940 Probleme, aber sie verstellten den Blick auf ein größeres Problem, daß wir – bis Ende 2018 - nämlich nur die Reproduzierbarkeit in der Theorie getestet haben, nicht aber in der Praxis.

Diesen Fokuswechsel, erst meinerseits, dann suksessive weitergegeben an andere Projektteilnehmer, betrachte ich persönlich als den größten Effekt dieses Prototypefund Projekts, da dies die weitere Arbeit entscheidend verändert hat.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Leider gibt es derzeit noch ein theoretisches und praktisches Grundproblem, d.h. leider kein Basissystem reproduzierbar ist, da bestimmte Pakete (perl, gcc-8, apt, sudo, groff, nss, xorg-server und python2.7 um nur 8 unreproduzierbare Pakete von 71 unreproduzierbaren Paketen der 1337 meist installierten Debian-Paketen zu nennen) nicht reproduzierbar sind. Dadurch ist es derzeit noch nicht möglich, ein (kleines) System zu bauen, das 100% reproduzierbar ist.

Aus diesem Grund habe ich an der praktischen Reproduzierbarkeit von diversen Systemen (Debian, Archlinux, OpenWrt) gearbeitet, bis ich an kurz- und mittelfristig unlösbare Probleme stieß und habe mich dann einem anderen System zugewandt, bis ich wieder irgendwo hängenblieb, so daß ich mir dann wieder ein anderes System vorgenommen habe.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Bis Ende März 2019 haben wir in 206 wöchentlichen Blogposts über die Arbeit unseres Projektes informiert (seit April 2019, also nach Projektende, schreiben wir monatliche Blogposts), diese finden sich unter https://reproducible-builds.org/reports/ wobei für das Prototypefund Projekt die Wochen 176-201 relevant sind.

https://reproducible-builds.org ist die generelle Webseite des Projekts, unsere Testergebnisse finden sich auf https://tests.reproducible-builds.org

Sourcecode findet sich unter https://salsa.debian.org/reproducible-builds/ und unter https://salsa.debian.org/qa/jenkins.debian.net/

https://www.youtube.com/watch?v=PSxm2DbDHG8 ist ein 90 Sekunden langes Video über meine Arbeit, produziert vom Prototypefund.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die Arbeitsplanung war sehr dynamisch, da kontinuierlich an der Reproduzierbarkeit von verschiedenen Projekten gearbeitet wurde und dies immer in Kooperation mit anderen Stakeholdern geschah. Dies hatte den Vorteil, daß es quasi keine Leerläufe gab, hatte allerdings auch den Nachteil, daß dadurch manche Arbeiten unterbrochen wurden (z.B. Archlinux) und erst nach Wochen oder Monaten wieder aufgenommen wurden.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Neben der Entwicklung und des Betriebs von https://tests.reproducible-builds.org besteht eine weitere Hauptrolle von mir innerhalb des Reproducible Builds Projekts darin, Anfragen und Arbeiten von beteiligten oder sich beteiligenden wollenden Projekten zu koordinieren und fördern, sowohl im IRC (Chat) als auch auf unseren Mailinglisten oder bei Konferenzen. Diese Arbeiten nehmen wöchentlich zwischen 1-10 Stunden (von mir) in Anspruch, die sich dann durch die Mitarbeit anderer teilweise extrem auszahlen. Zum Beispiel wurde Herr Alexander Couzens durch mich intensiv motiviert, diverse Reproduzierbarkeitsprobleme in OpenWrt zu lösen.

Mitte Dezember 2018 erhielt ich eine Email von OSUOSL (Oregon State University Open Source Labs), daß wir Zugriff auf acht Server mit *jeweils* 144 GiB RAM und 32 Cores bekommen würden (zusätzlich zu den bestehenden 44 Testsystemen). Die Einbindung dieser Server in unser existierende Test-Setups war dann eine unerwartetete Aufgabe.

Im September 2018 wurde außerdem unser Logo fertiggestellt, welches dann an diversen Stellen eingebunden werden mußte.

Der Reproducible Builds Summit in Paris vom 11.-13. Dezember, an dem 50 Leute aus 25 Projekten teilgenommen haben, hat sowohl in der Vor- als auch Nachbereitung einige Zeit gekostet, wobei die Synergieeffekte dieses Treffens beachtlich sind und den Zeitaufwand mehr als wettmachen.