Archive rebuilds, on your desk

Emanuele Rocca
MiniDebConf Cambridge 2026

What happens if we build all of Debian with…

GCC 16? glibc 2.43? New build flags?

@ema@fosstodon.org - CC BY-SA 4.0

arm

Outline

Why run archive-wide experiments?

Defining an experiment

Running an experiment

What we found

Future Work and Conclusions

@ema@fosstodon.org - CC BY-SA 4.0

arm

Why run archive-wide experiments?

arm

Archive-wide experiments

Change one thing, rebuild everything, see what breaks

- New toolchains: glibc, GCC, binutils, …
- New default build flags (see *Packages not using the default build flags: a taxonomy*, DebConf25)
- Higher ISA baseline, eg. `-march=armv9-a`

@ema@fosstodon.org - CC BY-SA 4.0

arm

Experiment: glibc 2.43

February 2026

- Rebuilt the archive against glibc 2.43
- Bugs filed with usertag `glibc-2.43` (user `debian-glibc@lists.debian.org`)
- Issues with 2MB THP (eg: #1128400)
- ISO C23 const return types (eg: #1128545)
- 98 bugs (86 done)

@ema@fosstodon.org - CC BY-SA 4.0

Experiment: GCC 16

March 2026

- Rebuilt the archive for the upgrade to GCC 16
- What is going to break once it becomes the default compiler?
- Bugs filed with usertag `ftbfs-gcc-16`
  (user `debian-gcc@lists.debian.org`)
- nodejs, protobuf, openmpi, jemalloc, libreoffice, ...
- 261 bugs (230 done)
- Set as default on August 1st

@ema@fosstodon.org - CC BY-SA 4.0

arm

Experiment: ARMv9 baseline

- Rebuilding Ubuntu Resolute with `-march=armv9-a`
- Internal compiler error in GCC 16: PR 127579
- Linux kernel build failure in a Marvell driver: patch merged in Linux v7.4
- Only built main: 1176 packages

@ema@fosstodon.org - CC BY-SA 4.0

Experiment and control

- Lots of packages FTBFS anyway
- Need a **control**: which failures are caused **by the experiment**?
- Retry all failures without the experiment conditions

@ema@fosstodon.org - CC BY-SA 4.0

Experiment and control

False Positives

- Flaky builds!
- What is worse than a package failing 100% of the times?
- A package failing 50% of the times.

@ema@fosstodon.org - CC BY-SA 4.0

Prior art

Lucas Nussbaum's archive rebuilds

Lucas' rebuilds

- Distributed: spawns VMs on AWS
- Ruby tooling, a big system: `collab-qa-tools`

My rebuilds

- Single host: your own hardware
- A handful of shell scripts

@ema@fosstodon.org - CC BY-SA 4.0

The Whole Thing

experiment.sh

```
systemd-run --user --scope --collect --quiet \
  --unit="sbuild-experiment-$(date +%s)-$(systemd-escape "$1")" \
  -p ManagedOOMMemoryPressure=auto -- \
  sbuild --append-to-version=$REBUILD_LABEL --arch-all --build-dir
    $BUILD_DIR \
  --apt-distupgrade \
  --chroot-setup-commands='printf "APPEND CFLAGS -O3\nAPPEND CXXFLAGS -O3\
    n" > /etc/dpkg/buildflags.conf' \
  --arch $ARCH --dist $DIST $1 > /dev/null || ./control.sh $1
```

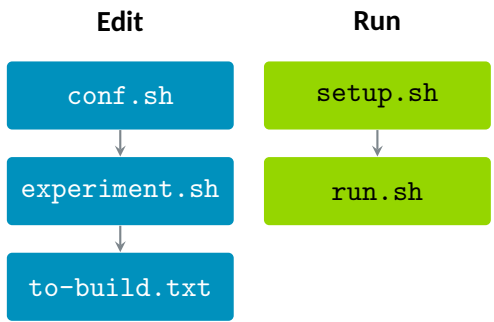

@ema@fosstodon.org - CC BY-SA 4.0

arm

Design

Simple parallel rebuild infrastructure: no daemon, no database, no root

- sbuild + mmdebstrap + GNU parallel + systemd-oomd
- An experiment: custom conf.sh and experiment.sh
- Package list in to-build.txt
- Run setup.sh to configure the environment



@ema@fosstodon.org - CC BY-SA 4.0

Sbuild

Custom sbuild.conf

```
$chroot_mode='unshare';
$unshare_tmpdir_template='/var/tmp/tmp.sbuild.XXXXXXXXXX';
$run_lintian=0;
$clean_source=0;
$purge_build_deps='always';
$purge_build_directory='always';
$purge_session='always';
```


@ema@fosstodon.org - CC BY-SA 4.0

Autopkgtests

- **From**: see if it builds
- **To**: see if it builds and kinda works too
- Add `$run_autopkgtest = 1;` to sbuild.conf!

@ema@fosstodon.org - CC BY-SA 4.0

Defining an experiment

Configuration

conf.sh example: Ubuntu Resolute armv9 rebuild

```
export DIST=resolute
export COMPONENTS="main universe"
export MIRROR=http://127.0.0.1:9999/ubuntu
export KEYRING=/usr/share/keyrings/ubuntu-archive-keyring.gpg

export REBUILD_LABEL=armv9
```

REBUILD_LABEL appended to package version, eg:
tcpdump_4.99.6-1armv9_arm64.deb

@ema@fosstodon.org - CC BY-SA 4.0

Setup

General idea

- In a fresh Debian VM, you only have to run setup.sh
- VM can be on your lap, on your desk, or in the cloud
- No additional configuration needed
- Start building immediately

@ema@fosstodon.org - CC BY-SA 4.0

Setup

setup.sh

- Script to configure the environment
- sbuild in **unshare** mode: no root needed, just subuid/subgid
- Chroot tarball: `mmdebstrap --mode unshare --variant=buildd`
- **approx** caches build dependencies locally
- Same scripts for Debian and Ubuntu: `conf.sh` sets `DIST`, `MIRROR`, `KEYRING`, `ARCH`

@ema@fosstodon.org - CC BY-SA 4.0

Experiment Definition

experiment.sh example: Ubuntu Resolute armv9 rebuild

Append `-march=armv9-a` to **CFLAGS** and **CXXFLAGS**:

```
--chroot-setup-commands='printf "APPEND CFLAGS -march=armv9-a\nAPPEND
    CXXFLAGS -march=armv9-a\n" > /etc/dpkg/buildflags.conf'
```


@ema@fosstodon.org - CC BY-SA 4.0

arm

Experiment vs control

experiment.sh and control.sh

```
# experiment.sh
sbuild --append-to-version=armv9 \
  --chroot-setup-commands='printf "APPEND CFLAGS -march=armv9-a [...]"' \
  $pkg || ./control.sh $pkg

# control.sh: only if the experiment failed, stock build
sbuild $pkg && echo $pkg >> regressions.log
```

Fails in the experiment, builds in the control: **regression**

@ema@fosstodon.org - CC BY-SA 4.0

Package List

to-build.txt example: all Debian sid packages that aren't Architecture: all

One package per line, by hand or from `allsidpkgs.sh`:

```
SOURCES='http://deb.debian.org/debian/dists/sid/main/source/Sources.xz'
curl -s "$SOURCES" |
    xzcat |
    awk '
        /^Package:/ {pkg=$2}
        /^Architecture:/ {$1=""; arch=$0}
        /^$/ && (arch != " all") { print pkg }
        ' | sort -u
```


@ema@fosstodon.org - CC BY-SA 4.0

Running an experiment

arm

Host selection

A big box!

- Ampere Altra 128 cores, 256G ram
- AWS Graviton4 c8g.12xlarge 48 cores, 96G ram (armv9)

@ema@fosstodon.org - CC BY-SA 4.0

arm

How long does it take?

Ballpark numbers

- Altra: ∼5500 packages in ∼8 hours
- Physical machine, fast I/O

@ema@fosstodon.org - CC BY-SA 4.0

arm

Running in parallel

run.sh

```
shuf to-build.txt | grep -v '^#' |
    time ionice -c 3 taskset -c 0-${CORES} \
        parallel --joblog $LOGFILE --jobs 75% ./experiment.sh
```

- `shuf`: spread the huge packages, no `gcc-14` next to `gcc-15`
- `taskset`: leave a couple of cores to the rest of the system
- `ionice -c 3`: idle I/O priority
- `--joblog`: runtime and exit status of each build

@ema@fosstodon.org - CC BY-SA 4.0

arm

The parallel joblog

armv9 rebuild data

Excerpt from rebuild-1788422500.log:

```
Seq Starttime      JobRuntime Exitval Command
161 1788439101.423   3287.671       0 ./experiment.sh bind9
 67 1788426924.589  16744.558       0 ./experiment.sh dovecot
158 1788438968.329   4982.460       0 ./experiment.sh vim
 65 1788426671.381  18997.532     137 ./experiment.sh openjdk-25
157 1788438907.627   8034.501       0 ./experiment.sh rustc-1.93
163 1788439122.959  10678.714       0 ./experiment.sh gcc-15-cross
131 1788432572.235  18955.117       0 ./experiment.sh gcc-16-cross
```

137: the control build was killed (OOM)

@ema@fosstodon.org - CC BY-SA 4.0

Not killing the machine

systemd-oomd to the rescue

```
systemd-run --user --scope --collect --quiet \
  --unit="sbuild-experiment-$(date +%s)-$(systemd-escape "$1")" \
  -p ManagedOOMMemoryPressure=auto -- sbuild ...
```

- First attempt: one scope around the whole `parallel` run
- Any OOM situation killed the **whole rebuild**
- Now: one scope per sbuild, only the hog dies (`openjdk-25`)
- Unit names: see what is running with `systemctl --user`

@ema@fosstodon.org - CC BY-SA 4.0

arm

Monitoring with systemd

```
# Journal of the experiment and control builds
journalctl --user -u 'sbuild-experiment-*'
journalctl --user -u 'sbuild-control-*'

# What is building right now in the control pass?
$ systemctl --user list-units 'sbuild-control-*' --no-legend
sbuild-control-1793288066-openjdk-25.scope loaded active running
  /usr/bin/sbuild --build-dir /work/ema/archive-rebuild/output-2
  --arch arm64 --dist unstable openjdk-25
```


@ema@fosstodon.org - CC BY-SA 4.0

Nice journal entries

One scope per build, for free from systemd

```
Started sbuild-experiment-1793287999-burgerspace.scope -
  /usr/bin/sbuild --append-to-version=armv9 --arch-all ...
sbuild-experiment-1793287999-burgerspace.scope:
  Consumed 1min 10.831s CPU time.
```

- Unit name: start timestamp and package
- **Started**: the exact sbuild command line that was run
- **Consumed**: CPU time of the build

@ema@fosstodon.org - CC BY-SA 4.0

Monitoring

What is going on?

- `summary.sh`: count of each sbuild `Status`:
- `failed.sh`, `unfinished.sh`: grep the build logs
- htop configured with CPU, memory, swap and disk I/O meters
- Stop at any time, `still_tobuild.sh` gives you the list of packages left

@ema@fosstodon.org - CC BY-SA 4.0

arm

summary.sh

Count of each sbuild Status

```
$ ./summary.sh
   1113 successful
     63 attempted
     31 given-back
     16 skipped
```

- `attempted`: the build ran and failed
- `given-back`: temporary problem, eg: cannot install build-deps
- `skipped`: intentionally not built, eg: architecture not supported

@ema@fosstodon.org - CC BY-SA 4.0

arm

What we found

arm

Catch: kernel oops

Archive rebuilds are great stress tests

```
Unable to handle kernel NULL pointer dereference
CPU: 9 UID: 100999 PID: 651757 Comm: systemd.test
Call trace:
 __aa_label_next_not_in_set+0x24/0x120 (P)
 aa_label_is_subset+0x48/0x88
 aa_unix_file_perm+0x248/0x990
```

- systemd test suite, running in an unshare'd sbuild (UID 100999)
- Crashed AppArmor: fix in the pipeline, hopefully Linux 7.4

@ema@fosstodon.org - CC BY-SA 4.0

Catch: GCC ICE

Archive rebuilds are great stress tests

- Internal compiler error in GCC 16 when building Linux with `-march=armv9-a`
- *“Preprocessed source stored into /tmp/cc6EtFzy.out file, please attach this to your bugreport.”*
- PR 127579

@ema@fosstodon.org - CC BY-SA 4.0

Catch: Linux kernel source bug

Archive rebuilds are great stress tests

- Linux kernel build failure in a Marvell driver
- `https://lore.kernel.org/lkml/20261005162137.71397-1-emanuele.rocca@arm.com/`
- Patch merged in Linux v7.4

@ema@fosstodon.org - CC BY-SA 4.0

arm

Rebuild logs

Published results

- `logs-to-upload.sh`: failing log from experiment, successful log from control
- glibc 2.43: `https://people.debian.org/~ema/glibc-2.43-rebuilds/`
- GCC 16: `https://people.debian.org/~ema/gcc-16-rebuilds/`
- `output-1`: failure logs (experiment)
- `output-2`: control logs

@ema@fosstodon.org - CC BY-SA 4.0

Future Work and Conclusions

arm

Future Work

The more you fool around, the more you find out

Ideas for other configurations to compare:

- Works in Testing but breaks in Unstable
- Works on Ubuntu but breaks in Unstable

@ema@fosstodon.org - CC BY-SA 4.0

arm

Future Work

Fool Around More

Experiment and control in two different steps: first run experiments, then reboot and run controls on failed builds.

- VM with an odd number of cores (literally: 3)
- Build environment: too many cores, not enough cores
- Too much memory, not enough memory
- Santiago Vila does a ton of work on these

@ema@fosstodon.org - CC BY-SA 4.0

Future Work

Fool Around More

More ideas for other configurations to compare:

- Latest RC kernel vs. latest stable
- Arm architectural features
- Different kernel command line tweaks
- Another architecture altogether!

@ema@fosstodon.org - CC BY-SA 4.0

Conclusions

- Large box with sbuild/unshare + parallel + systemd is all you need
- Anyone can run an archive-wide experiment

What experiment would you run?

`https://salsa.debian.org/ema/debelf-tracker/-/tree/master/archive-rebuilds`

@ema@fosstodon.org - CC BY-SA 4.0

arm

Example: ISO C23

#1128545: gcl FTBFS with glibc 2.43

```
../h/../h/protoize.h:106:35: error: expected identifier
  or '(' before '_Generic'
extern void *bsearch (const void *key, const void *base, ...
```

- glibc 2.43 adds ISO C23 const-preserving macros for `bsearch`, `memchr`, `strchr`, ...
- Breaks gcl's own `protoize.h`. Fix: drop its `bsearch` prototype

@ema@fosstodon.org - CC BY-SA 4.0

Example: unused variables

abyss: FTBFS with GCC 16

```
Consensus.cpp:379:18: error: variable 'numIgnored'
  set but not used [-Werror=unused-but-set-variable=]
  379 |         unsigned numIgnored = 0;
      |                  ^~~~~~~~~~
cc1plus: all warnings being treated as errors
```

- GCC 16 is stricter about unused-but-set variables
- With `-Werror`: FTBFS

@ema@fosstodon.org - CC BY-SA 4.0

arm