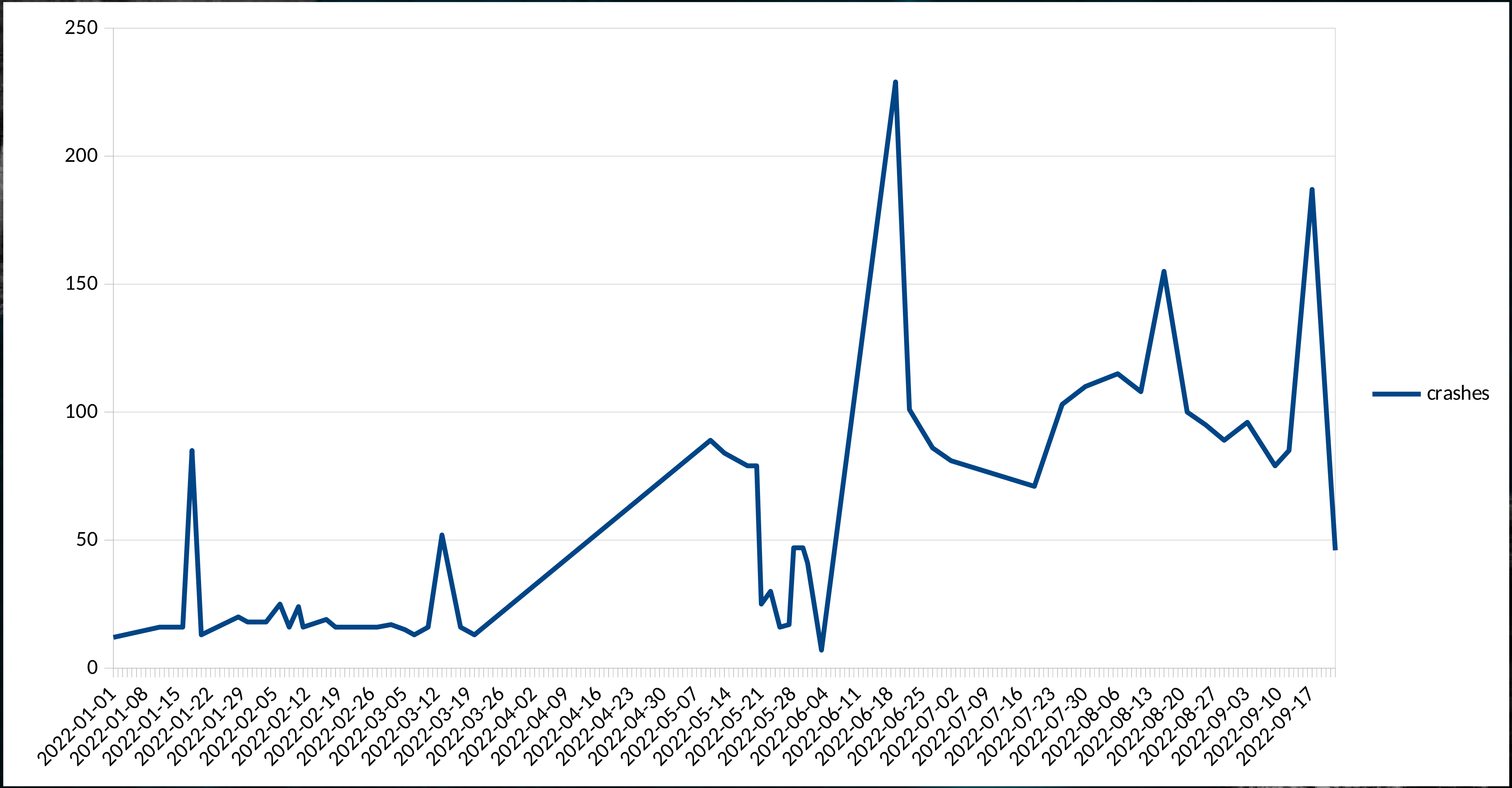
LibreOffice: crashtesting ⁊ fuzzing

CrashTesting

Overview

- Document Corpus
  - a) scraped out of various bugzilla instances with get-bugzilla-attachments-by-mimetype
  - b) scraped out of various online forums with get-forum-attachments.py
    - Thanks Xisco
  - c) some other donated collections, like cloudon and forcepoint
  - 264,953 files (116,200 in 2020)
- Import them all
  - With Markus Mohrhard’s test-bugzilla-files
- For many formats, then export to multiple formats
- Reimport exported output
- Report failed imports/exports
- Backtraces extracted from coredumps

2022 Crash Testing



- Big jump when new forum documents added

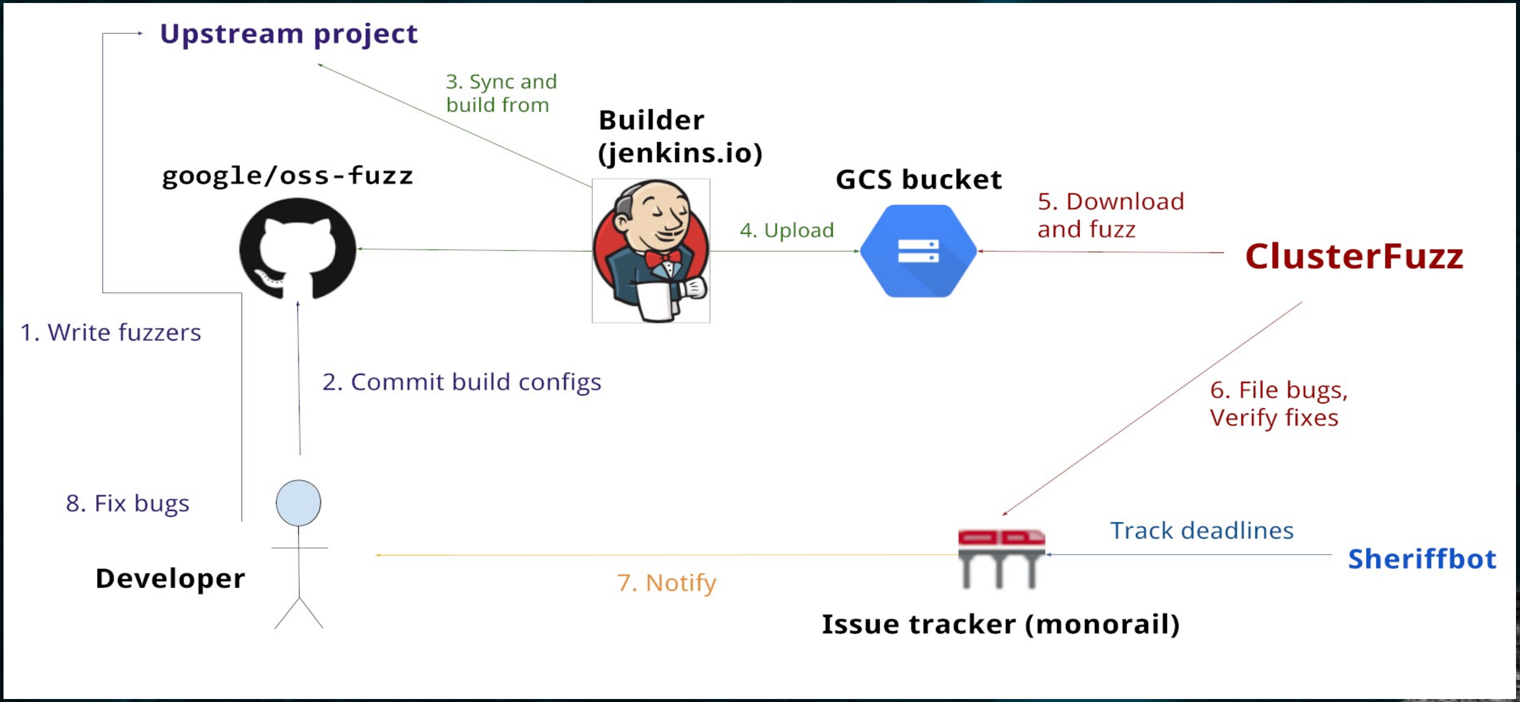
Thanks

- Thanks to Adfinis for the hardware
  - 48 cores, 128G memory → ~3 days
- Thanks to all the devs who have helped out to fix the findings
  - allotropia, Collabora and Red Hat
- Help wanted
  - 9 asserts/crashes outstanding
  - See mailing list report for details

Oss-Fuzz

Overview

- Continuous Fuzzing of our import filters
- Thanks to Google we get to use their infrastructure and resources



Configuration

- Build remotely on google’s side
  - Calls our bin/oss-fuzz-build.sh
  - Built twice a day
- 48 fuzzer targets in vcl/workben
- Each one is built with
  - libfuzzer + asan
  - libfuzzer + ubsan
  - libfuzzer + msan
  - honggfuzz + asan

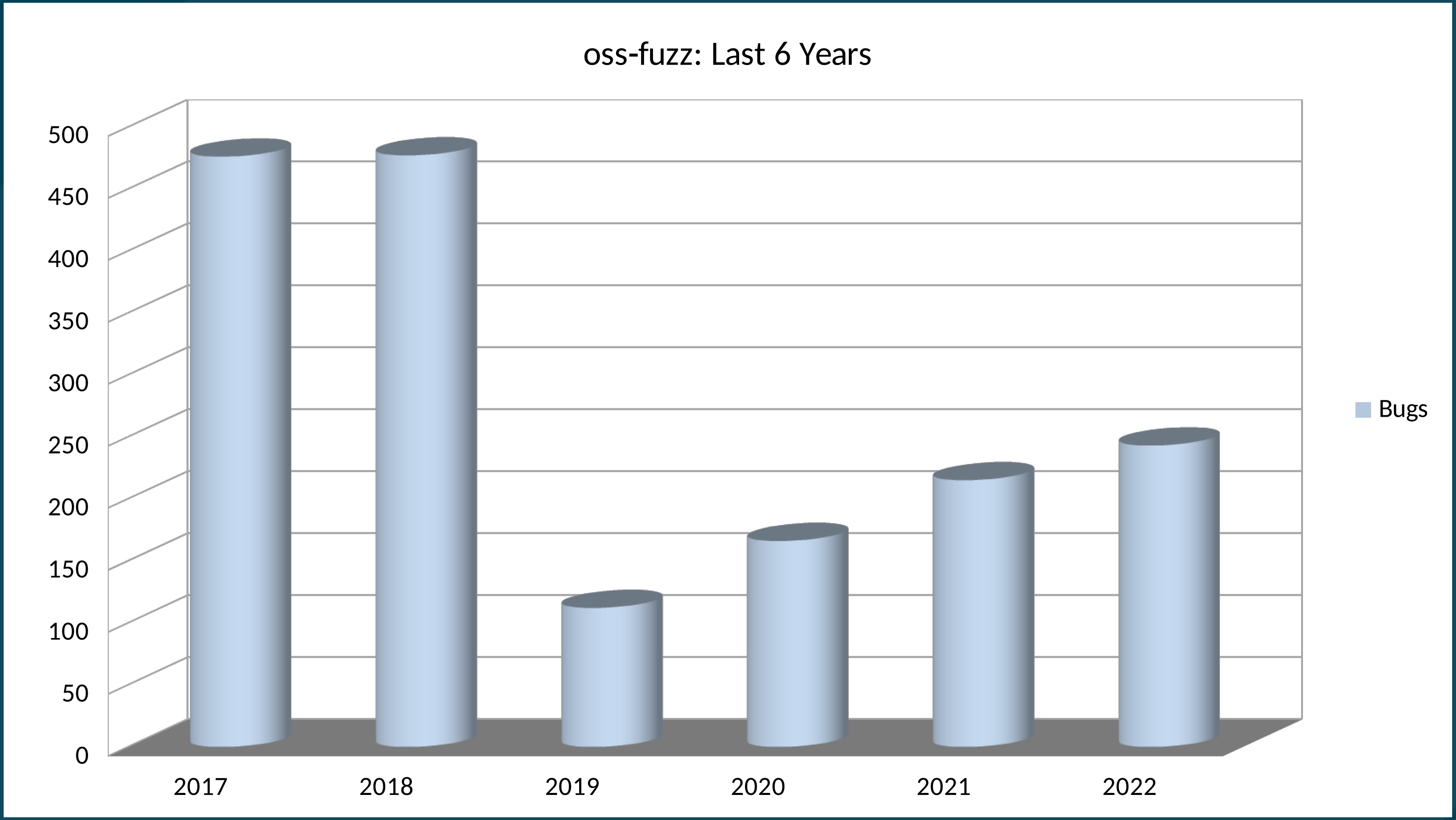
  => 192 total

Configuration

- Built statically
  - distro-configs/LibreOfficeOssFuzz.conf
  - Reuse --disable-dynamic-loading intended for iOS
  - Individual fuzzers are unfortunately v. large
- Run without config layer
  - Hardcoded suitable default for –enable-fuzzers
  - utl::ConfigManager::IsAvoidConfig()
- https://dev-www.libreoffice.org/corpus/
  - Contains our seed corpuses for 60 file formats
  - 15 are dtardons and co’s dlplib filters and are fuzzed separately

Oss-Fuzz Reports per Year

- Over 1768 issues over six years



- 243 this year to date
- 5 open at time of writing

Timeouts

- Sometimes timeout is genuine infinite loop
  - More often it’s just slow
- OssFuzz will report a maximum of one timeout per fuzzer
- Fix a timeout, another typically gets reported soon after
- Limit input size with a .options files

```
[libfuzzer]
max_len = 65536
```

- Some file formats have ~infinite decompression support
  - Tiny input can legitimately provide mega data to process
  - Examine FUZZ_MAX_INPUT_LEN (from .options) at runtime and limit to some factor of that

OOM

- Limit memory usage with

```
setenv("JPEGMEM", "768M", 1);
setenv("SC_MAX_MATRIX_ELEMENTS", "60000000", 1);
setenv("SC_NO_THREADED_CALCULATION", "1", 1);
```

- Pre-allocating buffers depending on potentially lying headers
  - Often a known relationship between remaining length of the file and the amount of data that it can produce
  - So short reads can be predicted before buffer allocation
    - GIF’s have a max compression of ~1:2560,
- Annoyingly, due to tracking freed memory?, ofz#49217 OOM persists though peak memory usage is far below limit, lots of buffers created and destroyed?

Coverage Gap

- We should fuzz exporting
- We should fuzz writing pdf
  - Which would exercise writer layout
  - Potentially challenging to deal with output