FOSDEM 22

Online Event, February 2022

Revisiting the Linux packaging for Apache OpenOffice

Analysis and future perspectives

Andrea Pescetti

pescetti@apache.org

Andrea Pescetti

- Active as a volunteer in several free and open source projects in my spare time.
- These include Apache OpenOffice, where I served as Project Chair and Release Manager.
- Responsible for Linux builds for a period.

The current packaging

Platforms and formats

OpenOffice binary packages

- Available as full binary packages in 41 languages, `ast` to `zh-TW`
- Covering (not counting ports) 4 platforms: Windows, OS X, Linux-64 and Linux-32
- Linux-64 and Linux-32 packages are available in two flavors: RPM and DEB, with the same content

A waste of space

- Apache OpenOffice downloads are in the hundreds of millions, so 1% is still huge
- That said, Linux packages take ~66% of the space for ~2% of downloads
- And that 2% is further unevenly distributed between Linux-64 (much more popular than Linux-32) and languages (top languages are orders of magnitude more popular than others), resulting in some combinations (niche platform in niche language) being practically never downloaded

System integration

- Our RPM and DEB packages are actually quite "universal"
- We build them on very old systems (baseline CentOS 5, released 2007) to ensure compatibility with virtually all Linux installations still alive today
- So this is a very generic packaging, not targeting any specific distribution and bundling lots of libraries rather than using the system ones

A different approach

The "universal" options available today

Three contenders

- Snap: developed by Ubuntu
- Flatpak: developed by Red Hat
- AppImage: independent "universal" technology

Common features

- Less burden on the distribution maintainers as these packages are supposed to be supplied by the vendor (think of an "app store" model); nothing changes for Apache OpenOffice as we are not maintained by most distributions anyway
- Much less integration with the Linux system; critics hate the duplication of libraries and the increased disk/memory footprint; but again, this is what OpenOffice already does
- Possibility to run versions in parallel

Snap

- Quite popular (~7000 packages) and seamlessly integrated in Ubuntu, command-line and GUI; supported by several other distributions too
- Even Firefox is currently offered only as a snap as of the latest Ubuntu 21.10; Ubuntu calls this the "deb-to-snap transition"
- It offers sandboxing for security and automatic updates; it uses a centralized store (Snapcraft); it has often been criticized for instability and poor performance

Flatpak

- Centralized store (Flathub) with ∼1500 packages; supported by several distributions, command-line and GUI
- Meant for desktop applications only
- It offers sandboxing for security and automatic updates

AppImage (best)

- Not a real "package" but a single executable file ("one application = one file"); just download, make executable and run
- Very few system requirements, perfect for the OpenOffice use case; updates require a full new download (but again, this is exactly what we are doing now)
- Fully decentralized, even if a "store" exists; fast and reliable; smallest file size; best in terms of application portability; no dependencies on corporations or Linux vendors

OpenOffice as AppImage

How to build it and how that could help

Building AppImages

1. Take an "installed" tree (either from the RPMs or the DEBs or any "installed" version like at the end of our build process)
2. Add a few configuration files
3. Run `appimagetool` for the final packaging

Tree structure

```
ApacheOpenOffice.AppDir/
├── AppRun
├── openoffice.desktop
├── openoffice.png
└── usr/
```

The AppRun file

```
ApacheOpenOffice.AppDir/
├── AppRun
├── openoffice.desktop
├── openoffice.png
└── usr/
```

as simple as

```
#!/bin/sh
export SAL_USE_VCLPLUGIN=gen
cd usr/program
./soffice
```

The .desktop file

```
ApacheOpenOffice.AppDir/
├── AppRun
├── openoffice.desktop
├── openoffice.png
└── usr/
```

as simple as

```
[Desktop Entry]
Name=ApacheOpenOffice
Exec=soffice
Icon=openoffice
Type=Application
Categories=Utility;
```

Other contents

```
ApacheOpenOffice.AppDir/
├── AppRun
├── openoffice.desktop
├── openoffice.png
└── usr/
```

- Our standard logo in PNG format
- `usr/` is a copy of `/opt/openoffice4` from any installation, or from the "installed" tree after build

Final build

- Get `appimagetool`, a very lightweight helper tool for the final packaging steps

```
cd ApacheOpenOffice.AppDir/
appimagetool-x86_64.AppImage --comp xz .
```

- An executable named `ApacheOpenOffice-x86_64.AppImage` is generated
- Distribute and enjoy!

Advantages

- Reduce disk usage for server (a 156M AppImage replaces 161M DEB + 164M RPM)
- Reduce disk usage for users (it just takes 156M, instead of ~400M needed by an installed version)
- Multiple versions readily available, that can be run together

Needing work/attention

- Investigate 32-bit support, which is very poor in all modern packaging technologies
- Incorporate this into our build system, as a final packaging target in addition to the current `RPM`, `DEB` and `installed`
- Avoid the "one more standard" joke! (XKCD#927: "There are 14 competing standards. Develop one universal standard that covers everyone's use cases. Then there are 15 competing standards.")

Thanks!

pescetti@apache.org